Digital
系列语音/传真设备
用户手册

（For Windows 2000/XP）

(Version Number 10.20)

深圳市数灵通电子有限公司

ShenZhen Digital Spirit Electronics CO., LTD.

2006 年 12 月 20 日

版权

本书所含信息以“维持原状”的方式提供，如有更改恕不另行通知。

本公司对本手册所含内容不做任何担保，包括但不限于暗示的适销性和特定用途适用性的担保。

本公司对本手册所含的错误或由于供应、执行或使用本手册或其中的示例而造成的意外或随发性损失不负责。

本公司致力于不断将新的技术运用于本公司的产品之中，本手册所列技术指标或配置如有更改，恕不另行通知。

本书含受版权保护的所有权信息。版权所有，翻版必究。除版权法允许外，未经本公司事先书面许可，不得对本书的任何部分进行复制、影印、再版、改编或翻译。

本产品中使用的程序版权所有。未经本公司事先书面许可，也不得复制、改编或反编译这些程序。

目 录

第一部分 DIGITAL 语音卡编程准备知识 5

第一章 DTMF 5
第二章 信号音 5
1.2.1 拨号音 5
1.2.2 回铃音 5
1.2.3 忙音 6
第三章 极性反转 6
第四章 语音以及语音压缩 6
1.4.1 A-率 PCM 码 6
1.4.2 ADPCM 码 6
第五章 主叫号码识别 7

第二部分 DIGITAL 语音卡软件编程说明 8

第一章 DIGITAL 语音卡编程架构说明 8
第二章 VC 平台编程范例 8
2.2.1 C 接口简介 8
2.2.2 C 语言编程简介 8
2.2.3 C 语言示范程序 9
第三章 VB 平台编程范例 9
2.3.1 VB 接口简介 9
2.3.2 VB 对动态库的调用 9
2.3.3 VB 编程简介 9
2.3.4 VB 编程的注意事项 9
2.3.5 VB 演示程序 10
第四章 C++ BUILDER 平台编程范例 10
2.4.1 C++ Builder 接口简介 10
2.4.2 C++ Builder 对动态库的调用 10
2.4.3 C++ Builder 下的示范程序 10
第五章 DELPHI 平台编程范例 10
2.5.1 Delphi 接口简介 10
2.5.2 Delphi 对动态库的调用 10
2.5.3 注意事项 10
2.5.4 Delphi 下的示范程序 11
第六章 PB 平台编程范例 11
2.6.1 PB 接口简介 11
2.6.2 PB 对动态库的调用 11
2.6.3 PB 下的示范程序 11
第七章 其他编程语言的调用方法 11

第三部分 DIGITAL 语音卡语音处理部分接口函数说明 12

第一章 DIGITAL 语音卡语音处理部分接口函数索引表（按功能分类） 12
第二章 DIGITAL 语音卡语音处理部分接口函数详细说明 17

3.2.1 初始化和关闭函数 17
3.2.2 获取系统参数接口函数 18
3.2.3 打开关闭通道接口函数 19
3.2.4 获取事件接口函数 20
3.2.5 侦听和连接通道接口函数 21
3.2.6 两线模拟中继模块接口函数 22
3.2.7 用户模块接口函数 23
3.2.8 基本录音接口函数 24
3.2.9 基本放音接口函数 26
3.2.10 DTMF 识别和产生接口函数 30
3.2.11 信号音产生接口函数 32
3.2.12 标准信号音识别接口函数 32
3.2.13 特殊信号音识别接口函数 37
3.2.14 话音/静音识别接口函数 37
3.2.15 超时设置接口函数 39
3.2.16 主叫识别和产生接口函数 39
3.2.17 发送 FSK 接口函数 41
3.2.18 自动拨号接口函数 42
3.2.19 语音数据格式转换接口函数 43
3.2.20 停止通道所有操作接口函数 44
3.2.21 会议接口函数 45
3.2.22 TTS(Text To Speech)接口函数 46
3.2.23 语音数据流处理函数 47
3.2.24 特殊语音格式的录音放音处理函数 47
3.2.25 特殊文件格式的录音放音处理函数 49
3.2.26 混音处理函数 52
3.2.27 点亮熄灭指示灯接口函数 53
3.2.28 读写用户信息接口函数 53

第四部分 DIGITAL 语音卡传真处理部分接口函数说明 55

第一章 DIGITAL 语音卡传真部分编程接口函数索引表（按功能分类） 55
第二章 DIGITAL 语音卡传真部分编程接口函数详细说明 56
4.2.1 获取系统参数接口函数 56
4.2.2 打开/关闭资源接口函数 57
4.2.3 绑定资源接口函数 57
4.2.4 传真资源操作接口函数 58

第五部分 DIGITAL 系列产品的技术支持 65

第一部分 Digital 语音卡编程准备知识

使用 Digital 语音卡编写应用程序需要了解通信方面的知识，这些也是必备的知识。下面介绍一些基本的电信常识。

第一章 DTMF

DTMF（Dual Tone Multi-Frequency），即双音多频信号，广泛用于电话拨号上，DTMF 由 CCITT 制定，并推荐作为按键式电话的标准。

每一个 DTMF 信号实际上是由两种音调的声音组合而成，一个是低频，另一个是高频。这样组成 DTMF 信号的频率就有两组，一组是低频信号 697Hz、770Hz、852Hz 和 941Hz，另一组是高频信号 1209Hz、1336Hz、1477Hz 和 1633Hz。

由四个低频信号和四个高频信号组合成 16 种 DTMF 信号——0~9 十个数字、*、#、A、B、C、D，这些信号的具体频率定义如下：

	1209Hz	1336Hz	1477Hz	1633Hz
697Hz	1	2	3	A
770Hz	4	5	6	B
852Hz	7	8	9	C
941Hz	*	0	#	D

在 Digital 语音卡上，DTMF 的识别和产生全部由 DSP 处理器实现。由于使用独特的算法，其识别准确度高于硬件识别。

第二章 信号音

在公共电话网上，线路的状态大都是通过信号音来表达的，如忙音、拨号音和回铃音等，下面介绍几种常用的信号音。

1.2.1 拨号音

公共电话网上的拨号音是在提机时，由交换机发出的一个 450Hz 的声音，一般它持续 6~10 秒，表示电话线路正常，并可以进行拨号呼叫。Digital 语音卡提供了 BF_StartDetectDialTone 函数来检测标准拨号音。

1.2.2 回铃音

公共电话网上的回铃音是在发出呼叫后，由交换机发出的一个 450Hz 的声音，表示被叫电话已经被接通，只要对方摘机，便可以进行对话。它是一个 1 秒有、4 秒无的有无间断的 450Hz 的声音。

Digital 语音卡提供了 BF_StartDetectBackTone 函数来检测标准回铃音。

1.2.3 忙音

忙音有几种形式，如在呼叫之后由于对方线路正忙而无法接通时，由交换机发出给主叫的信号音；也有在呼叫双方通话完毕，因一方挂机，而交换机给另一方发出的信号音。虽然国际规定忙音为 700 毫秒有、700 毫秒无的有无间隔的声音，但各个交换机厂家也有自己的忙音标准，致使忙音种类很多，需要用户小心使用。

第三章 极性反转

极性反转是在拨号之后，由交换机送出的用于识别对方是否摘机的信号。

判断对方摘机的一般方法是：当拨号之后，检测是否有回铃音，如果有回铃音再检测是否有语音，以判定对方是否摘机。但这种判断方法存在误差，而且无法分辨因对方占线或误拨等情况下由交换机送出的机器声音。

基于以上原因，使用极性反转来判断对方是否摘机是最准确的。但在使用本功能时，必须首先向电信公司申请电话的极性反转功能，因为该项功能并不是电信业务的基本功能，所以必须特殊申请。

Digital 语音卡既支持由语音判断对方摘机的功能，也支持用极性反转来判定对方摘机的功能。

注意：只有向电信局申请了极性反转功能，并且电信局的交换机支持极性反转功能，才能有效使用 Digital 语音卡为极性反转功能而提供的相应函数。

第四章 语音以及语音压缩

1.4.1 A-率 PCM 码

Pulse Code Modulation(PCM)是一种非均匀量化模拟信号的方法，该方法被广泛应用于电信业的语音传输和存储上。PCM 码是在模拟信号量化的基础上，再根据语音的特点，进行非均匀压缩。CCITT 的建议 G.711 规定了两种 PCM 的量化和压缩方法——A-律 PCM 码和 u-律 PCM 码。

u-律 PCM 为北美和日本所采用的语音压扩方法；而中国和欧洲采用 A-律 PCM 语音压扩方法。

经过A-律 PCM 编码后的模拟语音数据就变成了一个离散的数码流。一个 PCM 码的取样值对应 8 个比特，整个数码流的速率为 8kHz×8bit=64kbit/s。

Digital 语音产品在录音时，支持从模拟语音信号到 A-律 PCM 的数字化过程，并以 A-律 PCM 数字语音码存储语音文件；同时在放音时，支持从 A-律 PCM 的数字语音码到模拟语音信号的扩张过程。

1.4.2 ADPCM 码

ADPCM(Adaptive Differential Pulse Code Modulation)是 CCITT 推荐的语音压缩方法。它是根据语音的特点而制定的有损压缩方法，在 ADPCM 中，实际存储的不是语音的采样值，而是这一次采样与上一次采样的差值。由于语音信号的变化是连续、平滑而且很慢的，所以采样差值的位数减少，从而所需的存储空间也减少，达到压缩的目的。

Digital 语音卡不需要额外增加硬件，由板卡上 DSP 来实时处理 ADPCM 码的压扩。

Digital 语音卡提供的是 CCITT 建议的 32kbps 标准的 ADPCM 码的录音和放音。

第五章 主叫号码识别

主叫识别信息传送及显示(Calling Identity Delivery，简称 CID)业务是向被叫电话用户提供的一种新的服务项目。是指在被叫用户终端设备上显示主叫号码、主叫用户姓名、呼叫日期、时间等主叫识别信息，并进行存储，以供用户查阅的一种服务项目。

在模拟电话线路上，实现主叫识别信息的传送方法有两种——DTMF 和 FSK。

DTMF 方式是在第一声振铃之前使用 DTMF 音频信号传递，其特点是发送速度慢、无效验，而且仅仅发送主叫电话号码。

FSK 方式是在第一声振铃和第二声振铃之间使用 FSK 传递，特点是发送速度快、有效验，并且可以发送除主叫号码以外的信息，如主叫姓名、呼叫日期、时间等。但要注意，如果在第一声振铃时马上摘机，便无法收到主叫信息。

FSK(Frequency Shift Keying)，即频移键控码，在电信上，主要被用为主叫信息识别的信号。使用 FSK 传送主叫信息的国际标准有两种——BELL 和 CCITT 的，其中：

BELL 建议标准：

逻辑 0 用 2200Hz±1%表示

逻辑 1 用 1200Hz±1%表示。

CCITT 建议标准：

逻辑 0 用 2100Hz±1%表示

逻辑 1 用 1300Hz±1%表示。

我国原邮电部规定的主叫信息传送采用 BELL 标准，即

逻辑 0 用 2200Hz±1%表示

逻辑 1 用 1200Hz±1%表示。

在 Digital 语音卡上，FSK 信号的识别由硬件完成，并且遵循我国原邮电部规定的标准。

注　意：只有向电信局申请了主叫识别信息功能，并且电信局的交换机支持主叫识别信息功能，才能有效使用 Digital 语音卡为主叫识别信息功能而提供的相应函数。

第二部分 Digital 语音卡软件编程说明

Digital 系列语音卡为 Windows 系统平台上的用户提供的编程接口为动态连接库 BfVoice.dll，该文件存放在相应的 Windows 目录中。在实际编程中，用户可以根据实际情况进行操作。

Digital 系列语音卡也为 Windows 系统平台上的用户提供了各种编程语言的演示程序，这些程序都以源代码方式分发，用户在编程时，可以参考。

第一章 Digital 语音卡编程架构说明

Digital 系列语音卡的编程架构，分为三个部分，第一部分是初始化，第二部分是事件处理，第三部分是程序结束处理。下面分别叙述：

第一部分：初始化部分

该部分函数包含 BF_InitializeSystem 、 BF_GetTotalCard 、 BF_GetTotalChannel 、 BF_GetChannelType、BF_OpenChannel 和 BF_OpenSpecialChannel 等函数。这些函数仅仅在应用程序初始化的时候，调用一次，以初始化 Digital 语音系统和各个通道等。

第二部分：事件处理

Digital 系列语音卡的编程架构采用流行的事件驱动方式进行编程，即用户应用程序需要创建一个线程，在这个线程中，不断的调用函数 BF_GetMessage（），如果有事件返回，就处理事件，然后继续调用该函数。

第三部分：程序结束处理

在应用程序退出前，应该调用 Digital 系列语音卡的相应函数，清理系统，并恢复电话线路的状态。这些函数有 BF_CloseChannel 和 BF_CloseSystem 等。

关于 Digital 系列语音卡在各种语言平台上的编程，下面章节有详细叙述，请参考。

第二章 VC 平台编程范例

2.2.1 C 接口简介

BfVoice.LIB 是 BfVoice.DLL 的 C 语言输入库，由 Microsoft Visual C 6.0 生成。在 VC 环境下开发，可以直接将 Digital.LIB 加入到工程文件中。还有 BfVoice.h 声明文件，这两个文件都存放在驱动程序的目录。

2.2.2 C 语言编程简介

在 Windows 2000/XP 下用 C 语言对 Digital 系列语音卡进行编程时，一般包括三部分。

(1) 初始化部分，包括初始化卡，得到通道数和通道类型等等，这部分在运行时调用一次即可。

(2) 结束部分，包括关闭语音卡，这部分在退出时调用。

(3) 处理部分，这部分对实时的情况进行处理，是一个循环过程。

2.2.3 C 语言示范程序

C语言示范程序在VC 6.0的环境下开发。所有源代码都在发行盘的Demo\VC6子目录下。

第三章 VB 平台编程范例

2.3.1 VB 接口简介

在 Visual Basic 中，对动态库 BfVoice.DLL 的调用是通过 BfVoice.BAS 来声明的，请参考 BfVoice.bas。

在本质上，VB 也是通过调用 BfVoice.DLL 来实现对 Digital 系列语音卡硬件的控制。

2.3.2 VB 对动态库的调用

在 VB 中，需要首先对 DLL 中的函数进行声明，然后才能对该函数进行调用。关于如何声明 DLL 中的函数，请参考 VB 中的 HELP 文件“Declare statement”。

2.3.3 VB 编程简介

同在 Windowa 2000/XP 下用 C 语言编程一样，VB 对录音卡的操作，也包括三部分：

(1) 初始化部分，包括初始化卡、得到通道数和通道类型等等，在 Form_Load 中运行。

(2) 结束部分，包括关闭语音卡，在 Form_Unload 中运行。

(3) 处理部分，这部分在 TIMER 中运行，或放在 Form_Activate 函数里，具体用法请参见示范程序。

2.3.4 VB 编程的注意事项

(1) 所有参数和返回值的字节长度要相等。

(2) 在 VC 中（32 位环境），BOOL 为 32 位，WORD 为 16 位，DWORD 为 32 位；

在 VB 中，相应的使用 Long（32 位）, Integer（16 位）, Long（32 位）

在 VB 6.0 中，变量类型 Integer 仍然是 16 位的。其范围是-32767～32767。VC 中的 int, long, DWORD, BOOL 在 VB5.0 中需用 32 位的 long 表示；而 VC 中的 WORD 在 VB5.0 中需用 Integer 表示。

(3) 如果在 C 语言中某函数的参数为字符串，并且在函数调用完毕后，有内容通过该参数返回，当 VB 使用该函数时，带入的变量必须是已经分配空间的。

(4) 如果在 Form_Activate 中处理 Digital 语音卡的事件，Form_Activate 应该是一个死循环，要调用 DoEvents，以使得其他 Windows 2000/XP 程序能正常工作。

2.3.5 VB 演示程序

VB 语言示范程序在 VB 6.0 的环境下开发。所有源代码都在发行盘的 Demo\VB6 子目录下。

第四章 C++ Builder 平台编程范例

2.4.1 C++ Builder 接口简介

在 C++ Builder 中，对动态库 BfVoice.DLL 的调用的声明可以直接使用 VC 语言的 BfVoice.H，但是由于 BfVoice.LIB 是在 VC 环境下生成的，该 LIB 无法直接用于 C++ BUILDER 工程文件的连接。C++ BUILDER 提供了一个从 DLL 中提取输入库 LIB 的工具软件 IMPLIB.EXE，你可用该软件生成一个 LIB，用于开发。

2.4.2 C++ Builder 对动态库的调用

Implib 使用方法如下：

implib BfVoice.LIB BfVoice.DLL

对于 IMPLIB.EXE 的详细使用方法和功能，请参考其自身的帮助。

将生成的 BfVoice.LIB 拷贝到开发程序需要的地方，并加入到应用程序的工程文件中即可。

2.4.3 C++ Builder 下的示范程序

C++ Builder 语言示范程序在 C++ Builder 5.5 的环境下开发。所有源代码都在发行盘的 Demo\BC5 子目录下。

第五章 Delphi 平台编程范例

2.5.1 Delphi 接口简介

在 Delphi 中，对动态库 BfVoice.DLL 的调用是通过 BfVoice.PAS 来声明的。

2.5.2 Delphi 对动态库的调用

同在 VB 中一样，在 Delphi 中也需要首先对 DLL 中的函数进行声明，然后才能对该函数进行调用。关于如何声明 DLL 中的函数，请参考 Delphi 中的 HELP 文件“external declarations。

对于 BfVoice.DLL 的使用，用户可以参考声明文件 BfVoice.PAS。

2.5.3 注意事项

在 C 语言中的 char * 类型，要声明为 PChar。

2.5.4 Delphi 下的示范程序

Delphi 语言示范程序在 Delphi 6.0 的环境下开发。所有源代码都在发行盘的 Demo\Delphi6 子目录下。

第六章 PB 平台编程范例

2.6.1 PB 接口简介

在 Power Builder 中， 对动态库 BfVoice.dll 的调用是通过在“Declare”菜单中的“Global External Function”来声明的。

2.6.2 PB 对动态库的调用

对于 BfVoice.DLL 的使用，用户可以参考演示程序。

2.6.3 PB 下的示范程序

PB 语言示范程序在 Power Builder 7.0 的环境下开发。所有源代码都在发行盘的 Demo\PB7 子目录下。

第七章 其他编程语言的调用方法

在 Windows 2000/XP 的 32 位编程环境中，许多编程语言都可以用各种方式来调用 DLL 动态库。这样，也就可以通过对动态库 BfVoice.dll 的调用来控制 Digital 系列语音卡的硬件。

由于各种编程语言的方法不尽相同，本说明书无法一一描述。请用户仔细阅读你所用的编程语言的说明。

第三部分 Digital 语音卡语音处理部分接口函数说明

这一部分介绍 Digital 系列语音卡的语音处理部分的编程接口函数。

第一章 Digital 语音卡语音处理部分接口函数索引表（按功能分类）

初始化和关闭系统接口函数包括：

```
unsigned short WINAPI BF_InitializeSystem(void);
unsigned short WINAPI BF_CloseSystem(void);
```

获取系统参数接口函数

```
unsigned short WINAPI BF_GetTotalCard(void);
unsigned short WINAPI BF_GetCardSerialNo(unsigned short CardNo, unsigned
        char *SerialNo);
unsigned short WINAPI BF_SetCardUserID(unsigned short CardNo,unsigned char
        *Password,unsigned short PasseordLength,unsigned char *UserID);
unsigned short WINAPI BF_GetCardUserID(unsigned short CardNo,unsigned char
        *Password,unsigned short PasseordLength,unsigned char *UserID);
unsigned short WINAPI BF_GetTotalChannel(void);
unsigned short WINAPI BF_GetChannelType(unsigned short ChannelNo);
```

打开关闭通道接口函数

```
unsigned short WINAPI BF_OpenChannel(unsigned short ChannelNo);
unsigned short WINAPI BF_OpenSpecialChannel(unsigned short ChannelNo);
unsigned short WINAPI BF_CloseChannel(unsigned short ChannelNo);
```

获取事件接口函数

```
unsigned short WINAPI BF_GetMessage(unsigned short WaitTime,
        PBF_MESSAGE_INFO pMessageBuffer);
```

侦听与连接通道接口函数

```
unsigned short WINAPI BF_ListenChannel(unsigned short ChannelNo, unsigned short
        SourceChannel);
unsigned short WINAPI BF_UnListenChannel(unsigned short ChannelNo);
unsigned short WINAPI BF_LinkChannel(unsigned short FirstChannelNo, unsigned
        short SecondChannelNo);
unsigned short WINAPI BF_UnLinkChannel(unsigned short FirstChannelNo, unsigned
        short SecondChannelNo);
```

两线模拟中继模块操作接口函数包括：

```
unsigned short WINAPI BF_TrunkHookOff(unsigned short ChannelNo);
unsigned short WINAPI BF_TrunkHookOn(unsigned short ChannelNo);
unsigned short WINAPI BF_StartDetectTrunkPolarity(unsigned short ChannelNo);
```

```
    unsigned short WINAPI BF_StopDetectTrunkPolarity(unsigned short ChannelNo);
    unsigned short WINAPI BF_StartTrunkPickUp(unsigned short ChannelNo,unsigned short
            TimeByMilliSecond);
    unsigned short WINAPI BF_StopTrunkPickUp(unsigned short ChannelNo);
```

用户模块操作接口函数

```
    unsigned short WINAPI BF_UserRingOn(unsigned short ChannelNo,unsigned short
            RingOnTime,unsigned short RingOffTime,unsigned short TotalTimes);
    unsigned short WINAPI BF_UserRingOff(unsigned short ChannelNo);
    unsigned short WINAPI BF_SetUserPickUpTime(unsigned short TimeByMilliSecond);
```

录音接口函数

```
    unsigned short WINAPI BF_StartRecordFile(unsigned short ChannelNo,char
            *FileName,unsigned long Offset,unsigned long Length,char StopChar);
    unsigned short WINAPI BF_StartRecordBuffer(unsigned short ChannelNo,char
            *Buffer,unsigned long Offset,unsigned long Length,char StopChar);
    unsigned short WINAPI BF_StartRecordCircleBuffer(unsigned short ChannelNo,char
            *Buffer,unsigned long TotalLength,char StopChar);
    unsigned short WINAPI BF_StopRecordVoice(unsigned short ChannelNo);
    unsigned short WINAPI BF_GetRecordLength(unsigned short ChannelNo, unsigned long
            *RecordLength);
```

放音接口函数

```
    unsigned short WINAPI BF_StartPlayFile(unsigned short ChannelNo,char
            *FileName,unsigned long Offset,unsigned long Length,char StopChar);
    unsigned short WINAPI BF_StartPlayBuffer(unsigned short ChannelNo,char
            *Buffer,unsigned long Offset,unsigned long Length,char StopChar);
    unsigned short WINAPI BF_InitPlayIndexFile(unsigned short ChannelNo);
    unsigned short WINAPI BF_AddPlayIndexFile(unsigned short ChannelNo, char
            *FileName);
    unsigned short WINAPI BF_StartPlayIndexFile(unsigned short ChannelNo,char
            StopChar);
    unsigned short WINAPI BF_InitPlayIndexBuffer(void);
    unsigned short WINAPI BF_AddPlayIndexBuffer(char *Buffer,unsigned long Length);
    unsigned short WINAPI BF_StartPlayIndexBuffer(unsigned short ChannelNo,unsigned
            short *IndexTable,unsigned short IndexCount,char StopChar);
    unsigned short WINAPI BF_StartPlayCircleBuffer(unsigned short ChannelNo,char
            *Buffer,unsigned long TotalLength,char StopChar);
    unsigned short WINAPI BF_StopPlayVoice(unsigned short ChannelNo);
    unsigned short WINAPI BF_GetPlayLength(unsigned short ChannelNo, unsigned long
            *PlayLength);
```

DTMF 按键产生和识别接口函数

```
    unsigned short WINAPI BF_SetGenerateDtmfTime(unsigned short ChannelNo, unsigned
            short OnTime, unsigned short OffTime);
    unsigned short WINAPI BF_GenerateDTMFString(unsigned short ChannelNo, char
            *DTMFString);
    unsigned short WINAPI BF_StartDetectDTMF(unsigned short ChannelNo, unsigned short
```

```
        SmartLevel);
    unsigned short WINAPI BF_StopDetectDTMF(unsigned short ChannelNo);
    unsigned short WINAPI BF_ClearDTMFBuffer(unsigned short ChannelNo);
    unsigned short WINAPI BF_GetDTMFKey(unsigned short ChannelNo,unsigned short
        DTMFCount,char *DTMFKey);
    unsigned short WINAPI BF_CheckHitDTMF(unsigned short ChannelNo);
```

信号音产生接口函数

```
    unsigned short WINAPI BF_StartGenerateSignal(unsigned short ChannelNo,unsigned
        short Frequency0,unsigned short Frequency1,unsigned short OnTime,
        unsigned short OffTime,unsigned short TotalTimes,char StopChar);
    unsigned short WINAPI BF_StopGenerateSignal(unsigned short ChannelNo);
```

标准信号音识别接口函数

```
    unsigned short WINAPI BF_SetDetectToneInfo(unsigned short ChannelNo,
        PBF_TONE_INFO ToneInfo);
    unsigned short WINAPI BF_SetDetectDialToneFrequency(unsigned short ChannelNo,
        unsigned short Frequency);
    unsigned short WINAPI BF_SetDetectBackToneFrequency(unsigned short ChannelNo,
        unsigned short Frequency);
    unsigned short WINAPI BF_SetDetectBusyToneFrequency(unsigned short ChannelNo,
        unsigned short Frequency);
    unsigned short WINAPI BF_SetDetectBusyTone2Parameter(unsigned short ChannelNo,
        unsigned short BusyToneMaxTotalTime, unsigned short
        BusyToneMinTotalTime,unsigned short BusyToneRatio,unsigned short
        BusyToneErrorBetweenTwo,unsigned short BusyToneTotalTimes);
    unsigned short WINAPI BF_SetDetectDualBusyToneFrequency(unsigned short
        ChannelNo, unsigned short Frequency1, unsigned short Frequency2);
    unsigned short WINAPI BF_StartDetectDialTone(unsigned short ChannelNo);
    unsigned short WINAPI BF_StopDetectDialTone(unsigned short ChannelNo);
    unsigned short WINAPI BF_StartDetectBackTone(unsigned short ChannelNo);
    unsigned short WINAPI BF_StartDetectBackToneStop(unsigned short ChannelNo,
        unsigned short StopTimeByMs);
    unsigned short WINAPI BF_StopDetectBackToneStop(unsigned short ChannelNo);
    unsigned short WINAPI BF_StopDetectBackTone(unsigned short ChannelNo);
    unsigned short WINAPI BF_StartDetectBusyTone(unsigned short ChannelNo);
    unsigned short WINAPI BF_StopDetectBusyTone(unsigned short ChannelNo);
    unsigned short WINAPI BF_StartDetectToneData(unsigned short ChannelNo);
    unsigned short WINAPI BF_StopDetectToneData(unsigned short ChannelNo);
```

特殊信号音识别接口函数

```
    unsigned short WINAPI BF_SetSpecialSignalFrequency(unsigned short ChannelNo,
        unsigned short Frequency);
    unsigned short WINAPI BF_StartDetectSpecialSignal(unsigned short ChannelNo,
        unsigned short TimeByMilliSecond);
    unsigned short WINAPI BF_StopDetectSpecialSignal(unsigned short ChannelNo);
```

语音/静音识别接口函数

```
    unsigned short WINAPI BF_StartDetectHelloVoice(unsigned short ChannelNo);
    unsigned short WINAPI BF_StopDetectHelloVoice(unsigned short ChannelNo);
    unsigned short WINAPI BF_SetDetectHelloVoiceThreshold(unsigned short ChannelNo,
            unsigned short ThresholdValue);
    unsigned short WINAPI BF_StartDetectSilence(unsigned short ChannelNo,unsigned
            short ThresholdValue);
    unsigned short WINAPI BF_StopDetectSilence(unsigned short ChannelNo);
```

超时设置接口函数

```
    unsigned short WINAPI BF_StartTimeOut(unsigned short ChannelNo, unsigned short
            TimeBySecond);
    unsigned short WINAPI BF_StopTimeOut(unsigned short ChannelNo);
    unsigned short WINAPI BF_SetTimeOutValue(unsigned short ChannelNo, unsigned short
            TimeBySecond);
```

主叫识别和产生接口函数

```
    unsigned short WINAPI BF_StartDetectCallerID(unsigned short ChannelNo);
    unsigned short WINAPI BF_StopDetectCallerID(unsigned short ChannelNo);
    unsigned short WINAPI BF_GetRawFskCallerID(unsigned short ChannelNo, char
            *CallerIDString);
    unsigned short WINAPI BF_GetFskCallerID(unsigned short ChannelNo,char
            *DateTime,char *TelNo,char *Name);
    unsigned short WINAPI BF_StartDetectDtmfCallerID(unsigned short ChannelNo);
    unsigned short WINAPI BF_StopDetectDtmfCallerID(unsigned short ChannelNo);
    unsigned short WINAPI BF_GetDtmfCallerID(unsigned short ChannelNo,char *TelNo);
    unsigned short WINAPI BF_SendFskCallerID(unsigned short ChannelNo,unsigned char
            *DateTime,unsigned char *TelNo);
```

发送 FSK 接口函数

```
    unsigned short WINAPI BF_StartSendFsk(unsigned short ChannelNo,unsigned short
            SeizureFlag,char *Buffer,unsigned short Count);
    unsigned short WINAPI BF_StopSendFsk(unsigned short ChannelNo);
```

自动拨号接口函数

```
    unsigned short WINAPI BF_AutoDialOutWithHello(unsigned short ChannelNo,char
            *DTMFString);
    unsigned short WINAPI BF_AutoDialOutWithPole(unsigned short ChannelNo,char
            *DTMFString);
    unsigned short WINAPI BF_AutoDialOutWithoutDialToneWithHello(unsigned short
            ChannelNo,unsigned short WaitTimeBySecond,char *DTMFString);
    unsigned short WINAPI BF_AutoDialOutWithHelloNoColorRing(unsigned short
            ChannelNo,char *DTMFString);
```

语音数据格式转换接口函数

```
    unsigned short WINAPI BF_AlawToWave(char *AlawFileName, char *WaveFileName);
    unsigned short WINAPI BF_WaveToAlaw(char *WaveFileName, char *AlawFileName);
    unsigned short WINAPI BF_AlawToAdpcm(char *AlawFileName, char *AdpcmFileName);
    unsigned short WINAPI BF_AdpcmToAlaw(char *AdpcmFileName, char *AlawFileName);
    unsigned short WINAPI BF_G711ToG726(char *G711FileName,char
```

```
        *G726FileName,unsigned short G726VoiceType);
    unsigned short WINAPI BF_G726ToG711(char *G726FileName,unsigned short
        G726VoiceType,char *G711FileName);
```

停止通道所有操作接口函数

```
    unsigned short WINAPI BF_StopChannelOperate(unsigned short ChannelNo);
```

会议接口函数

```
    unsigned short WINAPI BF_ConferenceAddMember(unsigned short ChannelNo,unsigned
        short Mode,unsigned short Group);
    unsigned short WINAPI BF_ConferenceDelMember(unsigned short ChannelNo,unsigned
        short Mode,unsigned short Group);
    unsigned short WINAPI BF_ConferenceAddMemberWithGain(unsigned short
        ChannelNo,unsigned short Group, short Gain);
```

TTS(Text To Speech)接口函数

```
    unsigned short WINAPI BF_InitTTS(void);
    unsigned short WINAPI BF_CloseTTS(void);
    unsigned short WINAPI BF_TTSFileToFile(char *TextFileName, char *VoiceFileName);
    unsigned short WINAPI BF_TTSBufferToBuffer(char *TextBuffer,unsigned long
        TextCount,char *VoiceBuffer,unsigned long *VoiceCount);
    unsigned short WINAPI BF_TTSBufferToFile(char *TextBuffer,unsigned long
        TextCount,char *VoiceFileName);
```

语音数据流处理函数

```
    unsigned short WINAPI BF_ReadVoicePacket(unsigned short ChannelNo, char *Buffer);
    unsigned short WINAPI BF_WriteVoicePacket(unsigned short ChannelNo, char
        *Buffer);
```

特殊语音格式的录音放音处理函数

```
    unsigned short WINAPI BF_SetVoiceType(unsigned short ChannelNo, unsigned short
        VoiceType);
    unsigned short WINAPI BF_SetPlayVoiceType(unsigned short ChannelNo, unsigned
        short VoiceType);
    unsigned short WINAPI BF_SetRecordVoiceType(unsigned short ChannelNo, unsigned
        short VoiceType);
    unsigned short WINAPI BF_StartPlayFileByVoiceType(unsigned short ChannelNo,char
        *FileName,unsigned long Offset,unsigned long Length,char StopChar);
    unsigned short WINAPI BF_StartRecordFileByVoiceType(unsigned short ChannelNo,
        char *FileName,unsigned long Offset,unsigned long Length,char StopChar);
```

特殊文件格式的录音放音处理函数

```
    unsigned short WINAPI BF_SetPlayVoiceFileType(unsigned short ChannelNo, unsigned
        short FileType);
    unsigned short WINAPI BF_SetRecordVoiceFileType(unsigned short ChannelNo,
        unsigned short FileType);
    unsigned short WINAPI BF_StartPlayFileByFileType(unsigned short ChannelNo,char
        *FileName,unsigned long Offset,unsigned long Length,char StopChar);
    unsigned short WINAPI BF_StartRecordFileByFileType(unsigned short ChannelNo,char
        *FileName,unsigned long Offset,unsigned long Length,char StopChar);
```

```
unsigned short WINAPI BF_StartPlayAdpcmFile(unsigned short ChannelNo,char
        *FileName,unsigned long Offset,unsigned long Length,char StopChar);
unsigned short WINAPI BF_StartRecordAdpcmFile(unsigned short ChannelNo,char
        *FileName,unsigned long Offset,unsigned long Length,char StopChar);
```

混音处理函数

```
unsigned short WINAPI BF_StartMixVoice(unsigned short FirstChannelNo,unsigned
        short SecondChannelNo);
unsigned short WINAPI BF_StopMixVoice(unsigned short FirstChannelNo,unsigned
        short SecondChannelNo);
unsigned short WINAPI BF_MixVoiceFile(char *FirstFileName, char *SecondFileName,
        char *TargetFileName, unsigned short MixFlag);
```

USB 系列语音卡专用函数

```
unsigned short WINAPI BF_ControlLed(unsigned short LedNo, unsigned
        short LightFlag);
unsigned short WINAPI BF_WriteUserDataToUSB(unsigned char Address,
        unsigned char Length, unsigned char *DataBuffer);
unsigned short WINAPI BF_ ReadUserDataFromUSB(unsigned char Address,
        unsigned char Length, unsigned char *DataBuffer);
```

第二章 Digital 语音卡语音处理部分接口函数详细说明

3.2.1 初始化和关闭函数

● BF_InitializeSystem

原　型：　unsigned short WINAPI BF_InitializeSystem(void);

功　能：　该函数初始化 Digital 语音卡的驱动程序参数，所以在使用任何 Digital 提供的接口函数前，一定要调用该函数

参　数：　无

返　回：　如果函数成功，返回 BF_OK；否则返回错误码。错误码如下：

代号	数值	含义
BF_OK	0x0000	没有错误
BF_ERROR	0xFFFF	一般性错误

注　释：　只有该函数返回 BF_OK，语音卡才能正常操作。如果该函数调用失败，请检查语音卡的硬件和软件设置

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_CloseSystem

原　型：　unsigned short WINAPI BF_CloseSystem(void);

功　能：　该函数释放 Digital 语音卡的驱动程序所申请的系统资源，所以当停止 Digital 语音卡操作时，应该调用该函数。

参　数：　无

返　回：　如果函数成功，返回 BF_OK；否则返回错误码。

注　释：　如果调用过该函数之后，需要再次使用语音卡，需要再次调用函数 BF_InitializeSystem 来初始化语音卡

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.2 获取系统参数接口函数

● BF_GetTotalCard

原　型：　unsigned short WINAPI BF_GetTotalCard(void);
功　能：　该函数取得插入计算机中的 Digital 语音卡的数量
参　数：　无
返　回：　函数返回插入计算机中的 Digital 语音卡的数量
注　释：　如果函数成功，返回 BF_OK
适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_GetCardSerialNo

原　型：　unsigned short WINAPI BF_GetCardSerialNo(unsigned short CardNo, unsigned char *SerialNo);
功　能：　该函数取得插入计算机中的指定语音卡的系列号。对于任一 Digital 语音卡，都有一个三十字节的系列号，用以识别 Digital 语音卡
参　数：　CardNo　　卡的逻辑号码，从 0 开始
　　SerialNo　　返回的卡的系列号的指针，该缓存区由用户提供
返　回：　如果函数成功，返回 BF_OK
注　释：　无
适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_SetCardUserID

原　型：　unsigned short WINAPI BF_SetCardUserID(unsigned short CardNo,　unsigned char *Password,unsigned short PasseordLength, unsigned char *UserID);
功　能：　该函数设置指定语音卡的用户 ID 和密码。
参　数：　CardNo　　卡的逻辑号码，从 0 开始
　　Password　　用户密码，长度 1--15
　　PasswordLength　用户密码长度，取值 1--15
　　UserID　　用户 ID，长度 1--15
返　回：　如果函数成功，返回 BF_OK
注　释：　为了保护用户的应用软件，Digital 语音卡为用户应用程序在每快板卡上提供了一个 16 个字节的加密区域，该区域允许用户依据自己的喜好，写入 ID 与 Password 长度共为 16 字节的加密数据，因为板卡依据 Password 给 ID 数据加密，所以 ID 与 Paaword 长度均不能为 0。
适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_GetCardUserID

原　型：　unsigned short WINAPI BF_GetCardUserID(unsigned short CardNo, unsigned char *Password,unsigned short PasseordLength, unsigned char *UserID);
功　能：　该函数根据用户密码获取指定语音卡的用户 ID。
参　数：　CardNo　　卡的逻辑号码，从 0 开始
　　Password　　用户密码，长度 1--15
　　PasswordLength　用户密码长度，取值 1--15
　　UserID　　获取回来的用户 ID，长度 1--15
返　回：　如果函数成功，返回 BF_OK
注　释：　为了保护用户的应用软件，Digital 语音卡为用户应用程序在每快板卡上提供了一个 16 个字节的加密区域，该区域允许用户依据自己的喜好，写入 ID 与 Password 长度共为 16 字节的加密数据，因为板卡依据 Password 给 ID 数据加密，所以 ID 与 Paaword 长度均不

能为 0。为了保护用户数据起见，板卡并不记录设置时的密码长度，所以此函数必须输入正确的密码长度。

适用性： BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_GetTotalChannel

原　型： unsigned short WINAPI BF_GetTotalChannel(void);

功　能： 该函数取得插入计算机中的 Digital 语音卡的语音逻辑通道数量

参　数： 无

返　回： 函数返回插入计算机中的 Digital 语音卡的语音逻辑通道数量

注　释： 逻辑通道数量仅仅包括语音通道的

适用性： BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_GetChannelType

原　型： unsigned short WINAPI BF_GetChannelType(unsigned short ChannelNo);

功　能： 该函数获取语音逻辑通道号码为 ChannelNo 的通道所对应的物理通道类型

参　数： ChannelNo　　语音逻辑通道号码

返　回： 函数返回语音逻辑通道号码为 ChannelNo 的通道所对应的物理通道类型

物理通道类型常量的定义如下：

代号	数值	含义
CHANNEL_TYPE_NONE	0	逻辑通道所对应的物理通道无任何模块
CHANNEL_TYPE_USER	1	逻辑通道所对应的物理通道为用户模块
CHANNEL_TYPE_TRUNK	2	逻辑通道所对应的物理通道为中继模块
CHANNEL_TYPE_RECORD	3	逻辑通道所对应的物理通道为录音模块

注　释： ChannelNo 不能超过最大逻辑通道数量减一

适用性： BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.3 打开关闭通道接口函数

● BF_OpenChannel

原　型： unsigned short WINAPI BF_OpenChannel(unsigned short ChannelNo);

功　能： 打开语音通道，本函数仅仅适用于语音通道

参　数： ChannelNo　要打开通道的逻辑号码

返　回： 如果打开成功，返回 BF_OK

注　释： 对于每一个逻辑通道，只有被打开之后才能正常工作。

(1) 对于内线模块，通道打开后，Digital 驱动程序给它所对应的物理通道供电，并实时检测内线模块的摘机、挂机和拍插动作，并在内线模块有动作时，传递相应事件给应用程序。

(2) 对于中继模块，通道打开后，Digital 驱动程序实时检测中继模块的振铃和极性信息，并在中继模块有振铃时，传递振铃事件；在模块摘机后，开始检测极性是否反转，在有极性反转时，发送极性反转事件，而在模块挂机后，停止极性反转检测，如果需要特殊的极性反转检测，请使用 BF_StartDetectTrunkPolarity 和 BF_StopDetectTrunkPolarity 函数。

(3) 对于录音模块，通道打开后，Digital 驱动程序实时检测与录音模块并接的话机的摘机和挂机动作，并在有动作时，传递相应事件给应用程序。

适用性： BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_OpenSpecialChannel

原　型：unsigned short WINAPI BF_OpenSpecialChannel(unsigned short ChannelNo);

功　能：该函数为特殊模块而设，所谓的特殊模块是指其既具有外线（中继）模块的功能，也具有录音模块的功能，换句话说，调用该函数而非 BF_OpenChannel 函数初始化的通道，如果是特殊模块，可能获取的事件有：

MESSAGE_TRUNK_RING、MESSAGE_TRUNK_POLE、
MESSAGE_TRUNK_PICK_UP_END、MESSAGE_RECORD_OPEN_CIRCLE、
MESSAGE_RECORD_NO_OPEN_CIRCLE、MESSAGE_RECORD_HOOK_OFF、
MESSAGE_RECORD_HOOK_ON 和 MESSAGE_RECORD_RING 等，

有时对同一动作，可能有两个事件产生，如振铃动作，就有 MESSAGE_TRUNK_RING 和 MESSAGE_RECORD_RING 两个事件，编程时只需处理必要的事件即可。

参　数：ChannelNo　通道的逻辑号码

返　回：调用成功，返回 BF_OK

注　释：特殊通道的类型可以是 CHANNEL_TYPE_TRUNK 或 CHANNEL_TYPE_SPECIAL。
在应用过程中，对于这种特殊模块，事件出现的情况如下：
(1) 如果并线在挂机状态有摘机动作，会有事件：
MESSAGE_TRUNK_RING 和 MESSAGE_RECORD_HOOK_OFF
(2) 如果并线在摘机状态有挂机动作，会有事件：
MESSAGE_RECORD_HOOK_ON
(3) 如果线上有振铃动作，会有事件：
MESSAGE_RECORD_RING 和 MESSAGE_TRUNK_RING
(4) 如果模块在挂机状态执行摘机动作，会有事件：
MESSAGE_RECORD_RING 和 MESSAGE_RECORD_HOOK_OFF
(5) 如果模块在摘机状态执行挂机动作，会有事件：
MESSAGE_RECORD_HOOK_ON

所以，在应用时应注意，在不同状态下，应当仅仅检测有关的事件，如：
(1) 在空闲状态下，应只检测事件 MESSAGE_RECORD_RING 和 MESSAGE_RECORD_HOOK_OFF。
(2)在模块执行摘机和挂机动作时，应该忽略所有 RECORD 事件。

适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_CloseChannel

原　型：unsigned short WINAPI BF_CloseChannel(unsigned short ChannelNo);

功　能：关闭语音通道，本函数仅仅适用于语音通道

参　数：ChannelNo　要关闭通道的逻辑号码

返　回：如果打开成功，返回 BF_OK

注　释：当不再使用已经打开的某些通道时，为减少语音卡资源的占用，可以关闭不再使用的通道。如果要使用某个已经被关闭的通道，必须再调用函数 BF_OpenChannel 重新打开。

适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.4 获取事件接口函数

● BF_GetMessage

原　型：unsigned short WINAPI BF_GetMessage(unsigned short WaitTime, PBF_MESSAGE_INFO pMessageBuffer);

功　能：获取 Digital 语音卡的事件，该函数将事件从事件队列中取出。

参　数：WaitTime　指示如果没有事件，该函数可以被挂起的时间，单位毫秒。
WaitTime 的取值可以从 0 至 0xFFFF，建议取值为 50。

如果 WaiTime 的取值为 0，则不管是否有事件，都立即返回；
如果 WaiTime 的取值为 0xFFFF，则挂起该调用，直到有事件才返回；
如果 WaiTime 的取值为 0x0001 至 0xFFFE 之间的任意一个值，则为挂起的时间数量，其单位为毫秒。

pMessageBuffer　指向 PBF_MESSAGE_INFO 结构的地址指针，用于存放返回的事件，PBF_MESSAGE_INFO 的结构定义如下：

```
typedef struct _BF_MESSAGE_INFO{
    USHORT MessageCode;
    USHORT ChannelNo;
    USHORT Parameter[6];
} BF_MESSAGE_INFO, *PBF_MESSAGE_INFO;
```

如果该函数返回 BF_OK，则表示有事件发生，事件类型以及其它指示存于上面结构中。PBF_MESSAGE_INFO 结构中的各项含义如下：

MessageCode 指示事件的类型，取值请参考 BfVoice.h 文件
ChannelNo　指示发生事件的逻辑通道号码。
Parameter　与相应事件相关联的参数，不同的事件附带不同的参数，关于事件类型及其它参数的详细解释，请参考事件类型的相关章节。

返　回：　如果有事件，返回 BF_OK，相应的事件内容存于 pMessageBuffer 中。

注　释：　一定在本函数返回 BF_OK 时，pMessageBuffer 中的内容才有意义。
用户应用程序需要不断的循环调用该函数，取得各个通道的事件，并进行处理或根据实际需要发送命令，才能保证 Digital 语音卡系统的正常运行。一般情况下，在初始化 Digital 语音卡和打开通道之后，便开始循环调用该函数。

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.5 侦听和连接通道接口函数

● BF_ListenChannel

原　型：　unsigned short WINAPI BF_ListenChannel(unsigned short ChannelNo, unsigned short SourceChannel);

功　能：　侦听通道，即逻辑通道号码为 ChannelNo 的通道侦听逻辑通道号码为 SourceChannel 的通道。

参　数：　ChannelNo　目的通道的逻辑号码
SourceChannel　源通道的逻辑号码

返　回：　如果调用成功，返回 BF_OK

注　释：　当函数成功调用后，目的通道的信息来源于被侦听的通道，而在这种情况下发送放音命令时，将无法正确放音。

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_UnListenChannel

原　型：　unsigned short WINAPI BF_UnListenChannel(unsigned short ChannelNo);

功　能：　停止侦听，即停止逻辑通道号码为 ChannelNo 的通道侦听其它通道。

参　数：　ChannelNo　目的通道的逻辑号码

返　回：　如果调用成功，返回 BF_OK

注　释：　无

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_LinkChannel

原　型：　unsigned short WINAPI BF_LinkChannel(unsigned short FirstChannelNo, unsigned short SecondChannelNo);

功　能：　连接两个通道，即 FirstChannelNo 的通道侦听 SecondChannelNo 的通道，SecondChannelNo 的通道侦听 FirstChannelNo 的通道

参　数：　FirstChannelNo　　通道的逻辑号码
SecondChannelNo　　通道的逻辑号码

返　回：　如果调用成功，返回 BF_OK

注　释：　无

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_UnLinkChannel

原　型：　unsigned short WINAPI BF_UnLinkChannel(unsigned short FirstChannelNo,unsigned short SecondChannelNo);

功　能：　拆除两个通道之间的连接

参　数：　FirstChannelNo　　通道的逻辑号码

返　回：　如果调用成功，返回 BF_OK

注　释：　无

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.6 两线模拟中继模块接口函数

● BF_TrunkHookOff

原　型：　unsigned short WINAPI BF_TrunkHookOff(unsigned short ChannelNo);

功　能：　设置中继模块摘机。

参　数：　ChannelNo　　逻辑通道号码。

返　回：　如果调用成功，返回 BF_OK。

注　释：　对于中继模块通道，当有振铃事件或需要向外拨号时，必须首先摘机，才能进行其它操作。如在有振铃事件时，先摘机，然后进行放音、录音或 DTMF 识别等操作。

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_TrunkHookOn

原　型：　unsigned short WINAPI BF_TrunkHookOn(unsigned short ChannelNo)

功　能：　设置中继模块挂机。

参　数：　ChannelNo　　逻辑通道号码。

返　回：　如果调用成功，返回 BF_OK

注　释：　对于中继模块通道，当双方通话完毕或检测到对方挂机时，应该挂机。

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartDetectTrunkPolarity

原　型：　unsigned short BF_StartDetectTrunkPolarity(unsigned short ChannelNo);

功　能：　开始检测中继模块的极性反转，该函数主要用于在特殊情况下，需要检测极性反转时，使用该函数如果检测到有极性反转，则送出事件 MESSAGE_TRUNK_HAVE_POLE，并停止检测极性反转；如果需要停止检测，可以调用 BF_StopDetectTrunkPolarity 函数。当用于呼出时，判定对方是否摘机的极性反转，系统自动检测，无须调用该函数。

参　数：　ChannelNo　　逻辑通道号码

返　回：　如果调用成功，返回 BF_OK

注　释：　在使用本功能时，必须首先向电信公司申请极性反转功能。

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StopDetectTrunkPolarity

原　型：unsigned short BF_StopDetectTrunkPolarity(unsigned short ChannelNo);
功　能：停止检测中继模块的极性反转。
参　数：ChannelNo　逻辑通道号码。
返　回：如果调用成功，返回 BF_OK
注　释：无
适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartTrunkPickUp

原　型：unsigned short BF_StartTrunkPickUp(unsigned short ChannelNo,unsigned short TimeByMilliSecond);
功　能：启动中继模块的拍插动作。该函数实现中继模块挂机，然后等待指定时间（TimeByMilliSecond）后再摘机，同时发送拍插完毕事件。
参　数：ChannelNo　逻辑通道号码。　TimeByMilliSecond 拍插时间，以毫秒为单位
返　回：如果调用成功，返回 BF_OK
注　释：无
适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StopTrunkPickUp

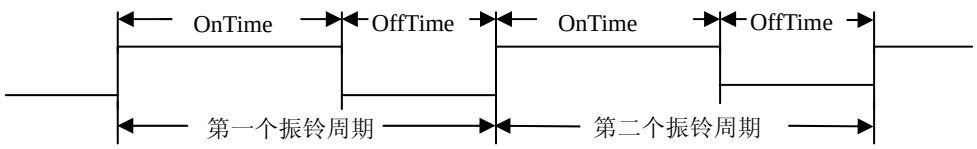
原　型：unsigned short BF_StopTrunkPickUp(unsigned short ChannelNo);
功　能：停止中继模块的拍插动作。该函数将立即将相应的中继模块摘机。
参　数：ChannelNo　逻辑通道号码。
返　回：如果调用成功，返回 BF_OK
注　释：无
适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.7 用户模块接口函数

● BF_UserRingOn

原　型：unsigned short BF_UserRingOn(unsigned short ChannelNo,unsigned short RingOnTime, unsigned short RingOffTime, unsigned short TotalTimes);
功　能：设置用户模块振铃。振铃信号是周期性的，每个周期由两段组成，即有无，每一段时间均可设定，总的振铃周期数量也可以设定。在振铃完毕时，会返回振铃结束(MESSAGE_USER_RING_END)事件，如果在振铃期间发送停止振铃命令，则立即停止振铃，并且没有任何事件返回。
参　数：ChannelNo　逻辑通道号码
RingOnTime　代表振铃时间，单位毫秒，取值 0x0000—0xFFFF。
RingOffTime　代表不振铃时间，单位毫秒，取值 0x0000—0xFFFF。
TotalTimes　代表总的振铃周期数量，取值 0x0001--0xFFFF。

送出的振铃信号波形如下：



返　回：　如果调用成功，返回 BF_OK

注　释：　无

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_UserRingOff

原　型：　unsigned short BF_UserRingOff(unsigned short ChannelNo);

功　能：　停止用户模块振铃。

参　数：　ChannelNo　逻辑通道号码。

返　回：　如果调用成功，返回 BF_OK

注　释：　无

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_SetUserPickUpTime

原　型：　BOOL unsigned short BF_SetUserPickUpTime(unsigned short TimeByMilliSecond);

功　能：　设置用户模块拍插时间。

参　数：　TimeByMilliSecond　以毫秒为单位的时间，该缺省值为 500ms。

返　回：　如果调用成功，返回 BF_OK

注　释：　该函数对整个系统的所有用户模块均发生作用，当用户模块有拍插动作时，Digital 语音卡发送 MESSAGE_USER_PICK_UP 事件。
所谓的拍插动作，就是当用户模块由摘机状态转换到挂机状态，然后再回到摘机状态的总共时间小于一定时间的动作。

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.8 基本录音接口函数

● BF_StartRecordFile

原　型：　unsigned short BF_StartRecordFile(unsigned short ChannelNo,char *FileName,unsigned long Offset,unsigned long Length,char StopChar);

功　能：　开始文件录音，将录音数据存于文件。可以设置录音长度，或设置在收到某个或某些 DTMF 按键时，停止录音。当录音数据达到要求时，停止录音，并送出事件 MESSAGE_RECORD_VOICE_END；当收到相应的 DTMF 按键时，停止录音，并送出事件 MESSAGE_RECORD_VOICE_EXIT。

参　数：　ChannelNo　通道的逻辑号码。

FileName　录音文件名称，含有文件存放的路径，最大长度为 MAX_FILE_NAME_NUM-1。

Offset　如果录音文件存在，存放语音的起始位置，如果录音文件不存在，必须设置为 0。

Length　需要录音数据的字节长度，如果设为 0xFFFFFFFF，则一直录音，直到发送录音停止命令，或有指定的 DTMF 收到。

StopChar　停止录音的 DTMF，当收到相应的 DTMF 按键时，停止录音，有效取值为：

0x00	表示收到任何 DTMF 按键都不停止
0x01	表示收到任一 DTMF 按键就停止
0—9,*,#,A,B,C,D	表示收到某个 DTMF 按键就停止
其他值	没有意义

返　回：　如果调用成功，返回 BF_OK

注　释：　如果要使 StopChar 生效，必须首先调用识别 DTMF 函数 BF_StartDetectDTMF

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartRecordBuffer

原　型：unsigned short BF_StartRecordBuffer(unsigned short ChannelNo,char *Buffer, unsigned long Offset,unsigned long Length,char StopChar);

功　能：开始缓存区录音，将录音数据存于缓存区。必须设置录音长度，可以设置在收到某个或某些 DTMF 按键时，停止录音。当录音数据达到要求时，停止录音，并送出事件 MESSAGE_RECORD_VOICE_END；当收到相应的 DTMF 按键时，停止录音，并送出事件 MESSAGE_RECORD_VOICE_EXIT。

参　数：

ChannelNo	通道的逻辑号码。	
FileName	录音文件名称，含有文件存放的路径，最大长度为 MAX_FILE_NAME_NUM-1。	
Offset	如果录音文件存在，存放语音的起始位置，如果录音文件不存在，必须设置为 0。	
Length	需要录音数据的字节长度，如果设为 0xFFFFFFFF，则一直录音，直到发送录音停止命令，或有指定的 DTMF 收到。	
StopChar	停止录音的 DTMF，当收到相应的 DTMF 按键时，停止录音，有效的取值为：	
	0x00	表示收到任何 DTMF 按键都不停止
	0x01	表示收到任一 DTMF 按键就停止
	0—9,*,#,A,B,C,D	表示收到某个 DTMF 按键就停止
	其他值	没有意义

返　回：如果调用成功，返回 BF_OK

注　释：如果要使 StopChar 生效，必须首先调用识别 DTMF 函数 BF_StartDetectDTMF

适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartRecordCircleBuffer

原　型：unsigned short BF_StartRecordCircleBuffer(unsigned short ChannelNo,char *Buffer,unsigned long TotalLength,char StopChar);

功　能：开始循环缓存区录音

参　数：

ChannelNo	通道的逻辑号码	
Buffer	缓存区指针	
TotalLength	循环缓存区的大小，单位为字节，其 最小取值为 0x400,必须成倍递增,如 0x400,0x800,0x1000 等，但不能为 0xC00。	
StopChar	指定如果通道收到某个或某些 DTMF 按键，停止录音，有效取值如下：	
	0x00	表示收到任何 DTMF 按键都不停止
	0x01	表示收到任一 DTMF 按键就停止
	0—9,*,#,A,B,C,D	表示收到某个 DTMF 按键就停止
	其他值	没有意义

返　回：如果调用成功，返回 BF_OK

注　释：循环缓存区录音是指 Digital 系统在给定的缓存区中，循环录音，在录音数据达到至整个缓存区后，再从缓存区的开始之处录音，并且在数据达到整个缓存　区或整个缓存区一半的时候，分别发送事件 MESSAGE_RECORD_CIRCLE_SECOND_END 和 MESSAGE_RECORD_CIRCLE_FIRST_END。

适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StopRecordVoice

原　型：unsigned short BF_StopRecordVoice(unsigned short ChannelNo);

功　能：停止录音，停止某个通道的录音操作 。

参　数：ChannelNo　通道的逻辑号码。

返　回：　如果调用成功，返回 BF_OK
注　释：　无
适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_GetRecordLength

原　型：　unsigned short BF_GetRecordLength(unsigned short ChannelNo, unsigned long *RecordLength);
功　能：　取得当前录音长度
参　数：　ChannelNo　　通道的逻辑号码
　　RecordLength　　存放当前录音长度
返　回：　如果调用成功，返回 BF_OK
注　释：　无
适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.9 基本放音接口函数

Digital 系列语音卡一系列功能强大的放音函数，既包括单一的文件或缓存区放音，也包括多文件或多缓存区的索引放音。

对于任何一种放音，如果需要中途停止放音，调用函数 BF_StopPlayVoice 即可。

(1) 对于文件索引放音：

Digital 系统为每一个逻辑通道独立建立一个最大长度为 64 的索引文件表，使用 BF_InitPlayIndexFile 初始化这个索引文件表，即将索引文件表清空；使用 BF_AddPlayIndexFile 增加项目到索引文件表；调用 BF_StartPlayIndexFile 函数进行索引文件放音，该函数从指定通道的索引文件表的第一个文件开始放音，当第一个文件结束时，自动放音下一个，直至索引表中的所有文件放音完毕，发送 MESSAGE_PLAY_VOICE_END 事件。

(2) 对于缓存区索引放音：

Digital 系统仅仅建立一个最大数量为 256 的缓存区索引表，所有通道的索引缓存区放音都使用相同的这个索引表，调用 BF_InitPlayIndexBuffer 函数来初始化这个索引表，即将索引表清空；调用 BF_AddPlayIndexBuffer 函数增加表中的项目，每个缓存区的加入是顺序进行的；调用 BF_StartPlayIndexBuffer 函数启动索引缓存区放音。

(3) 对于索引文件放音的应用，举例如下：

如有 0、1、2、年、月、日的语音文件，文件名分别为：Zero.tel、One.tel、Two.tel、Year.tel、Month.tel、Day.tel，那么要在第八通道放音“2001 年 2 月 1 日”这段话，只需如此操作：

```
BF_InitPlayIndexFile(8);
BF_AddPlayIndexFile(8,“Two.tel”);
BF_AddPlayIndexFile(8,“Zero.tel”);
BF_AddPlayIndexFile(8,“Zero.tel”);
BF_AddPlayIndexFile(8,“One.tel”);
BF_AddPlayIndexFile(8,“Year.tel”);
BF_AddPlayIndexFile(8,“Two.tel”);
BF_AddPlayIndexFile(8,“Month.tel”);
BF_AddPlayIndexFile(8,“One.tel”);
BF_AddPlayIndexFile(8,“Day.tel”);
BF_StartPlayIndexFile(8, StopChar);
```

(4) 对于索引文件放音的应用，举例如下：

应用系统首先申请足够数量的缓存区，然后将频繁使用的语音读入相应的缓存区，再建立索引表，即可使用索引缓存区放音。

如有 0、1、2、年、月、日的语音，缓存区的指针分别为：Zero、One、Two、Year、Month、Day，那么可以如下建立缓存区索引表：

```
BF_InitPlayIndexBuffer();
BF_AddPlayIndexBuffer(Zero, xxxx);
BF_AddPlayIndexBuffer(One, xxxx);
BF_AddPlayIndexBuffer(Two, xxxx);
BF_AddPlayIndexBuffer(Year, xxxx);
BF_AddPlayIndexBuffer(Month, xxxx);
BF_AddPlayIndexBuffer(Day, xxxx);
```

其中 xxxx 代表相应缓存区字节长度

那么，无论以后哪个通道进行索引缓存区放音，均使用这个索引表，如要在第八通道放音“2001 年 2 月 1 日”这段话，只需如此操作：

```
unsigned short Buffertable[9]={2,0,0,1,3,2,4,1,5};
BF_StartPlayIndexBuffer(8,BufferTable,9,StopChar);
```

其中 BufferTable 项的值，就是各个语音缓存区在缓存区索引表中的顺序号。

对于循环缓存区放音，是指 Digital 系统在给定的缓存区中，循环放音，在放音数据达到至整个缓存区后，再从缓存区的开始处放音，并且在数据达到整个缓存区或整个缓存区一半的时候，分别发送事件 MESSAGE_PLAY_CIRCLE_SECOND_END 和 MESSAGE_PLAY_CIRCLE_FIRST_END。

下面详细叙述各种放音相关函数：

● BF_StartPlayFile

原　型：unsigned short BF_StartPlayFile(unsigned short ChannelNo,char*FileName, unsigned long Offset,unsigned long Length,char StopChar);

功　能：开始文件放音，放音数据存于文件。当发送停止放音命令，或收到某个或某些设定的停止 DTMF 按键时，停止放音。当所有要求的放音数据放完时，停止放音，并送出事件 MESSAGE_PLAY_END；当收到相应的 DTMF 按键时，停止放音，并送出事件 MESSAGE_PLAY_EXIT。

参　数：

ChannelNo	通道的逻辑号码。	
FileName	如果放音数据在文件中，则指定文件名称，含有文件存放的路径，最大长度　为 MAX_FILE_NAME_NUM-1	
Ofsset	指定放音文件的字节偏移长度	
Length	指定放音文件的字节长度，如果设为 0xFFFFFFFF，则表示一直到文件结尾	
StopChar	停止放音的 DTMF，当收到相应的 DTMF 按键时，停止放音，有效的取值为：	
	0x00	表示收到任何 DTMF 按键都不停止
	0x01	表示收到任一 DTMF 按键就停止
	0—9,*,#,A,B,C,D	表示收到某个 DTMF 按键就停止
	其他值	没有意义

返　回：如果调用成功，返回 BF_OK

注　释：如果要使 StopChar 生效，必须首先调用识别 DTMF 函数 BF_StartDetectDTMF

适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartPlayBuffer

原　型：unsigned short BF_StartPlayBuffer(unsigned short ChannelNo,char*Buffer, unsigned long Offset,unsigned long Length,char StopChar);

功　能：开始缓存区放音，放音数据存于缓存区。当发送停止放音命令，或收到某个或某些设定的停止 DTMF 按键时，停止放音。当所有要求的放音数据放完时，停止放音，并送出事件 MESSAGE_PLAY_END；当收到相应的 DTMF 按键时，停止放音，并送出事件 MESSAGE_PLAY_EXIT。

参　数：

ChannelNo	通道的逻辑号码。	
Buffer	放音数据缓存区。	
Ofsset	指定放音文件的字节偏移长度	
Length	指定放音文件的字节长度，一定要设置，以防越界	
StopChar	停止放音的 DTMF，当收到相应的 DTMF 按键时，停止放音，有效的取值为：	
	0x00	表示收到任何 DTMF 按键都不停止
	0x01	表示收到任一 DTMF 按键就停止
	0—9,*,#,A,B,C,D	表示收到某个 DTMF 按键就停止
	其他值	没有意义

返　回：如果调用成功，返回 BF_OK

注　释：如果要使 StopChar 生效，必须首先调用识别 DTMF 函数 BF_StartDetectDTMF

适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_InitPlayIndexFile

原　型：unsigned short BF_InitPlayIndexFile(unsigned short ChannelNo)

功　能：初始化索引放音文件，该函数清空指定通道的索引放音。

参　数：ChannelNo　通道的逻辑号码。

返　回：如果调用成功，返回 BF_OK

注　释：无

适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_AddPlayIndexFile

原　型：unsigned short BF_AddPlayIndexFile(unsigned short ChannelNo, char *FileName);

功　能：向指定通道内增加索引文件

参　数：ChannelNo　通道的逻辑号码

返　回：如果调用成功，返回 BF_OK

注　释：可以增加的最大索引文件数量为 1024 个。

适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartPlayIndexFile

原　型：unsigned short BF_StartPlayIndexFile(unsigned short ChannelNo,char StopChar);

功　能：开始索引文件放音

参　数：

ChannelNo	通道的逻辑号码。	
StopChar	停止放音的 DTMF，当收到相应的 DTMF 按键时，停止放音，有效的取值为：	
	0x00	表示收到任何 DTMF 按键都不停止
	0x01	表示收到任一 DTMF 按键就停止
	0—9,*,#,A,B,C,D	表示收到某个 DTMF 按键就停止
	其他值	没有意义

返　回：如果调用成功，返回 BF_OK

注　释：　如果需要停止索引文件放音，调用 BF_StopPlayVoice 函数

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_InitPlayIndexBuffer

原　型：　unsigned short BF_InitPlayIndexBuffer(void);

功　能：　初始化缓存区索引，即将索引放音的缓存区清空

参　数：　无

返　回：　如果调用成功，返回 BF_OK

注　释：　无

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_AddPlayIndexBuffer

原　型：　unsigned short BF_AddPlayIndexBuffer(char *Buffer,unsigned long Length);

功　能：　添加索引缓存区

参　数：　Buffer　　缓存区
　　Length　　缓存区长度

返　回：　如果调用成功，返回 BF_OK

注　释：　可以增加的最大索引缓存区数量为 1024 个。

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartPlayIndexBuffer

原　型：　unsigned short BF_StartPlayIndexBuffer(unsigned short ChannelNo, unsigned short *IndexTable, unsigned short IndexCount,char StopChar);

功　能：　开始索引缓存区放音

参　数：　ChannelNo　通道的逻辑号码。
　　IndexTable　索引表，该索引表实际上是一个数组，该数组的每一个项指代缓存区索引表 中的相应缓存区的顺序号。
　　IndexCount　索引表中的项数
　　StopChar　停止放音的 DTMF，当收到相应的 DTMF 按键时，停止放音，有效的取值为：
　　0x00　表示收到任何 DTMF 按键都不停止
　　0x01　表示收到任一 DTMF 按键就停止
　　0—9,*,#,A,B,C,D　表示收到某个 DTMF 按键就停止
　　其他值　没有意义

返　回：　如果调用成功，返回 BF_OK

注　释：　如果需要停止索引文件放音，调用 BF_StopPlayVoice 函数

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartPlayCircleBuffer

原　型：　unsigned short BF_StartPlayCircleBuffer(unsigned short ChannelNo,char *Buffer,unsigned long TotalLength,char StopChar);

功　能：　开始循环缓存区放音

参　数：　ChannelNo　通道的逻辑号码
　　Buffer　缓存区指针
　　TotalLength　循环缓存区的大小，单位为字节，其 最小取值为 0x400,必须成倍递增,如 0x400,0x800,0x1000 等，但不能为 0xC00。
　　StopChar　指定如果通道收到某个或某些 DTMF 按键，停止放音，有效取值如下：

0x00	表示收到任何 DTMF 按键都不停止
0x01	表示收到任一 DTMF 按键就停止
0—9,*,#,A,B,C,D	表示收到某个 DTMF 按键就停止
其他值	没有意义

返　回：　如果调用成功，返回 BF_OK

注　释：　无

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StopPlayVoice

原　型：　unsigned short BF_StopPlayVoice(unsigned short ChannelNo);

功　能：　停止放音操作

参　数：　ChannelNo　通道的逻辑号码

返　回：　如果调用成功，返回 BF_OK

注　释：　所有的放音（如文件、缓存区、索引文件和索引缓存区）停止，都调用该函数。

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_GetPlayLength

原　型：　unsigned short BF_GetPlayLength(unsigned short ChannelNo, unsigned long *PlayLength);

功　能：　取得当前实际放音长度

参　数：　ChannelNo　通道的逻辑号码。

PlayLength　存放实际放音长度

返　回：　如果调用成功，返回 BF_OK

注　释：　无

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.10 DTMF 识别和产生接口函数

● BF_StartDetectDTMF

原　型：　unsigned short BF_StartDetectDTMF(unsigned short ChannelNo, unsigned short SmartLevel);

功　能：　开始识别 DTMF 按键，在识别到 DTMF 时，发送事件(MESSAGE_HAVE_DTMF)，并继续识别，直到发送停止识别命令。应用程序可以随时调用函数 BF_GetDTMFKey 来获取已经识别到的 DTMF 按键

参　数：　ChannelNo　通道的逻辑号码。

SmartLevel　指定识别 DTMF 的灵敏度，取值为 0-4，数字越大越灵敏，但应注意，灵敏度越大，识别的准确率可能降低。通常取中间值 2

返　回：　如果调用成功，返回 BF_OK

注　释：　一般情况下，灵敏度要取缺省值 2，特殊情况下，可以灵活取值

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StopDetectDTMF

原　型：　unsigned short BF_StopDetectDTMF(unsigned short ChannelNo);

功　能：　停止检测 DTMF 按键

参　数：　ChannelNo　通道的逻辑号码

返　回：　如果调用成功，返回 BF_OK

注　释：　无

适用性： BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_ClearDTMFBuffer

原　型： unsigned short BF_ClearDTMFBuffer(unsigned short ChannelNo);

功　能： 清除某一通道已经识别出来的但仍然没有被取走的 DTMF 按键

参　数： ChannelNo　通道的逻辑号码

返　回： 如果调用成功，返回 BF_OK

注　释： 该函数多数用在已经调用开始识别 DTMF 按键后，在获取新的按键之前，清除通道内残留的没有用的 DTMF 按键

适用性： BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_GetDTMFKey

原　型： unsigned short BF_GetDTMFKey(unsigned short ChannelNo, unsigned short DTMFCount,char *DTMFKey);

功　能： 获取 DTMF 按键，从 ChannelNo 通道内提取数量为 DTMFCount 个按键。如果含有足够数量的 DTMF 按键，则将按键依次添入 DTMFKey 所指的缓存区中，并返回 BF_OK；否则返回其他错误码。所有 DTMF 按键均为 ASCII 码，字母为大写

参　数： ChannelNo　通道的逻辑号码。
DTMFKey　存放取回的 DTMF 按键，数量为 DTMFCount。
DTMFCount　要取回的 DTMF 按键数量，取值为 1 ~ (MAX_DTMF_NUM - 1)

返　回： 如果含有所需数目的 DTMF 按键，返回 BF_OK，按键存于 DTMFKey 中

注　释： 无

适用性： BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_SetGenerateDtmfTime

原　型： unsigned short WINAPI BF_SetGenerateDtmfTime(unsigned short ChannelNo, unsigned short OnTime, unsigned short OffTime);

功　能： 设置发送 DTMF 双音频信号的时间长度和间隔长度

参　数： ChannelNo　通道的逻辑号码。
OnTime　发送双音频信号的时间长度，单位毫秒。缺省设置为 100 毫秒。
OffTime　发送两个双音频信号之间的间隔时间长度，单位毫秒。缺省值为 100 毫秒

返　回： 如果调用成功，返回 BF_OK

注　释： 无

适用性： BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_GenerateDTMFString

原　型： unsigned short BF_GenerateDTMFString(unsigned short ChannelNo, char *DTMFString);

功　能： 产生 DTMF 按键，时间参数可以由函数 BF_SetGenerateDtmfTime 设定，缺省设置为 100 毫秒有、100 毫秒无

参　数： ChannelNo　通道的逻辑号码。
DTMFString　要产生的 DTMF 按键的字符串，以 0x00 结尾，数量小于 MAX_DTMF_NUM-1。有效的 ASCII 字符为：”0”—“9”、”*”、”#”、”A”、“a”、”B”、”b”、”C”、”c”、”D”、”d” 、” ，”

返　回： 如果调用成功，返回 BF_OK

注　释： 符号“,”表示延时 500ms

适用性： BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.11 信号音产生接口函数

● BF_StartGenerateSignal

原　型：unsigned short BF_StartGenerateSignal(unsigned short ChannelNo,unsigned short Frequency0, unsigned short Frequency1, unsigned short OnTime, unsignedshort OffTime,unsigned short TotalTimes, char StopChar);

功　能：产生 300Hz—3400Hz 之间的一个或两个频率的有无间断的周期性声音，每一个周期分为两段，第一段产生信号音，第二段不产生信号音。并在产生完毕时，发送事件（MESSAGE_GENERATE_SIGNAL_END）；如果有指定的停止发送信号音的 DTMF 按键，则在收到指定的停止发送信号音的 DTMF 按键时，发送事件（MESSAGE_GENERATE_SIGNAL_EXIT），然后发送有 DTMF 事件；如果在产生信号音中间发送停止发送信号音命令，则立即停止发送信号音，并且没有任何事件产生

参　数：ChannelNo　通道的逻辑号码。

Frequency0　指定第一个频率的频率值，单位赫兹（Hz）。

Frequency1　指定第二个频率的频率值，单位赫兹（Hz）。

OnTime　指定产生信号音的时间，单位毫秒，取值 0x0000-0x7FFF。

OffTime　指定不产生信号音的时间，单位毫秒，取值 0x0000-0x7FFF。

TotalTimes　指定产生信号音的周期次数。

StopChar　指定如果通道收到某个或某些 DTMF 按键，停止产生信号音，有效值如下：

0x00	表示收到任何 DTMF 按键都不停止
0x01	表示收到任一 DTMF 按键就停止
0—9,*,#,A,B,C,D	表示收到某个 DTMF 按键就停止
其他值	没有意义

返　回：如果调用成功，返回 BF_OK

注　释：如果要停止产生信号音，必须调用函数 BF_StopGenerateSignal。如果要使 StopChar 生效，必须首先调用识别 DTMF 函数 BF_StartDetectDTMF

适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StopGenerateSignal

原　型：unsigned short BF_StopGenerateSignal(unsigned short ChannelNo);

功　能：停止产生信号音

参　数：ChannelNo　通道的逻辑号码

返　回：如果调用成功，返回 BF_OK

注　释：无

适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.12 标准信号音识别接口函数

● BF_SetDetectToneInfo

原　型：unsigned short BF_SetDetectToneInfo (unsigned short ChannelNo,PBF_TONE_INFO ToneInfo);

功　能：设置标准信号音的识别参数。一般情况下，使用缺省设置，不需要调用该函数。

参　数：ChannelNo　通道的逻辑号码。

ToneInfo　指向 PBF_TONE_INFO 结构的地址指针，用于存放标准信号音的各种参数，PBF_TONE_INFO 的结构定义如下：

```
typedef struct _BF_TONE_INFO{
    unsigned short   DialToneMinTotalTime;
    unsigned short   BusyToneMaxTotalTime;
    unsigned short   BusyToneMinTotalTime;
```

```
    unsigned short BusyToneRatio;
    unsigned short   BusyToneErrorBetweenTwo;
    unsigned short BusyToneTotalTimes;
    unsigned short   BackToneMaxTotalTime;
    unsigned short   BackToneMinTotalTime;
    unsigned short   BackToneRatio;
    unsigned short   BackToneErrorBetweenTwo;
    unsigned short   BackToneTotalTimes;
} BF_TONE_INFO, *PBF_TONE_INFO;
```

PBF_TONE_INFO 结构中的各项含义如下：

代号	含义	缺省值
DialToneMinTotalTime	拨号音持续最小时间(毫秒)	2500
BusyToneMaxTotalTime	最大忙音周期时间(毫秒)	1500
BusyToneMinTotalTime	最小忙音周期时间(毫秒)	500
BusyToneRatio	忙音周期占空比	50
BusyToneErrorBetweenTwo	两个忙音周期的误差比	10
BusyToneTotalTimes	忙音周期数量	3
BackToneMaxTotalTime	最大回铃音周期时间(毫秒)	6000
BackToneMinTotalTime	最小回铃音周期时间(毫秒)	4000
BackToneRatio	回铃音周期占空比	20
BackToneErrorBetweenTwo	两个回铃音周期的误差比	10
BackToneTotalTimes	回铃音周期数量	2

返　回：　如果调用成功，返回 BF_OK

注　释：　对于忙音参数，仅仅设置了一个频率的时间等识别参数，第二个忙音的识别参数的设置，请参阅函数 BF_SetDetectDualBusyToneFrequency 和 BF_SetDetectBusyTone2Parameter

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_SetDetectDialToneFrequency

原　型：　unsigned short BF_SetDetectDialToneFrequency (unsigned short ChannelNo, unsigned short Frequency);

功　能：　设置识别拨号音的频率。在调用该函数后，对于指定的通道，调用函数 BF_StatartDetectDialTone 所识别的频率，就是这个函数设定的频率。没有调用该函数时的缺省频率为 450 赫兹

参　数：　ChannelNo　　通道的逻辑号码。
　　Frequency　　频率值，单位为赫兹，取值范围为 300--3500

返　回：　如果调用成功，返回 BF_OK

注　释：　无

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_SetDetectBackToneFrequency

原　型：　unsigned short BF_SetDetectBackToneFrequency (unsigned short ChannelNo, unsigned short Frequency);

功　能：　设置识别回铃音的频率。在调用该函数后，对于指定的通道，调用函数 BF_StartDetectBackTone 所识别的频率，就是这个函数设定的频率。没有调用该函数时的缺省频率为 450 赫兹

参　数：　ChannelNo　　通道的逻辑号码。
　　Frequency　　频率值，单位为赫兹，取值范围为 300--3500

返　回：　如果调用成功，返回 BF_OK

注　释：　无

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_SetDetectBusyToneFrequency

原　型：unsigned short BF_SetDetectBusyToneFrequency (unsigned short ChannelNo, unsigned short Frequency);

功　能：设置识别单个忙音的频率。在调用该函数后，对于指定的通道，调用函数 BF_StartDetectBusyTone 和 BF_StartDetectToneData 所识别的频率，均为这个函数设定的频率。没有调用该函数时的缺省频率为 450 赫兹

参　数：ChannelNo　通道的逻辑号码。
Frequency　频率值，单位为赫兹，取值范围为 300--3500

返　回：如果调用成功，返回 BF_OK

注　释：无

适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_SetDetectDualBusyToneFrequency

原　型：unsigned short WINAPI BF_SetDetectDualBusyToneFrequency(unsignedshort ChannelNo, unsigned short Frequency1, unsigned short Frequency2);

功　能：设置识别忙音的频率，该函数设置两个忙音频率。调用该函数后，如果调用函数 BF_StartDetectBusyTone，系统将按照函数 BF_SetDetectToneInfo 设置的时间等参数，识别第一个频率；而按照函数 BF_SetDetectBusyTone2Parameter 设置的时间等参数识别第二个频率(两个频率的识别时间等参数可以不相同)，任一频率符合条件，均送出忙音事件 MESSAGE_HAVE_BUSY_TONE，同时停止识别忙音检测。同时函数 BF_StartDetectToneData 所识别的频率为第一个频率

参　数：ChannelNo　通道的逻辑号码。
Frequency1　第一个忙音频率，单位 Hz。
Frequency2　第二个忙音频率，单位 Hz

返　回：如果调用成功，返回 BF_OK

注　释：只要调用了该函数，系统就按照两个忙音参数进行忙音检测,所以必须在调用函数 BF_StartDetectBusyTone 之前调用函数 SetDetectBusyTone2Parameter 设置第二个忙音识别时间等参数。如果在调用改函数之后，想仅仅识别一个忙音频率，需要调用设置一个识别频率的函数 BF_SetDetectBusyToneFrequency

适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_SetDetectBusyTone2Parameter

原　型：unsigned short WINAPI BF_SetDetectBusyTone2Parameter(unsigned short ChannelNo, unsigned short BusyToneMaxTotalTime, unsigned shortBusyToneMinTotalTime, unsigned short BusyToneRatio, unsigned short BusyToneErrorBetweenTwo, unsigned short BusyToneTotalTimes);

功　能：设置识别第二个忙音的参数，该函数设置第二个忙音识别参数。该函数必须在函数 BF_StartDetectBusyTone 之前调用

参　数：ChannelNo　通道的逻辑号码。
BusyToneMaxTotalTime　最大忙音周期时间
BusyToneMinTotalTime　最小忙音周期时间
BusyToneRatio　忙音周期占空比
BusyToneErrorBetweenTwo　两个忙音周期的误差比
BusyToneTotalTimes　忙音周期数量

返　回：如果调用成功，返回 BF_OK

注　释：无

适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartDetectDialTone

原　型：unsigned short BF_StartDetectDialTone(unsigned short ChannelNo);

功　能：　开始识别拨号音，当识别到拨号音时，发送 MESSAGE_HAVE_DIAL_TONE(有拨号音)事件给应用程序。该信号音的频率可以调用函数 BF_SetDetectDialToneFrequency 设定，缺省值为 450Hz

参　数：　ChannelNo　通道的逻辑号码

返　回：　如果调用成功，返回 BF_OK

注　释：　该函数仅仅识别标准的拨号音，这个函数常常在中继模块的摘机之后，而在拨号之前调用，以验证线路是否可以拨号。在检测到标准拨号音后，Digital 语音卡继续检测标准拨号音，并不停止；如果要停止检测，必须调用函数 BF_StopDetectDialTone

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StopDetectDialTone

原　型：　unsigned short BF_StopDetectDialTone(unsigned short ChannelNo);

功　能：　停止检测标准拨号音

参　数：　ChannelNo　通道的逻辑号码

返　回：　如果调用成功，返回 BF_OK

注　释：　无

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartDetectBackTone

原　型：　unsigned short BF_StartDetectBackTone(unsigned short ChannelNo);

功　能：　开始识别标准的回铃音，当识别到回铃音时，发送 MESSAGE_HAVE_BACK_TONE (有回铃音)事件给应用程序。该信号音的频率可以调用函数 BF_SetDetectBackToneFrequency 设定，缺省值为 450Hz。

参　数：　ChannelNo　通道的逻辑号码

返　回：　如果调用成功，返回 BF_OK

注　释：　该函数仅仅识别标准的回铃音，这个函数常常在中继模块的拨号之后调用，以验证线路是否接通。在检测到标准回铃音后，Digital 语音卡继续检测标准回铃音，并不停止；如果要停止检测，必须调用函数 BF_StopDetectBackTone

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StopDetectBackTone

原　型：　unsigned short BF_StopDetectBackTone(unsigned short ChannelNo);

功　能：　停止检测标准回铃音

参　数：　ChannelNo　通道的逻辑号码

返　回：　如果调用成功，返回 BF_OK

注　释：　无

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartDetectBackToneStop

原　型：　unsigned short BF_StartDetectBackToneStop(unsigned short ChannelNo,unsigned short StopTimeByMs);

功　能：　开始识别标准的回铃音是否停止，当在 StopTimeByMs 时间内没有识别到回铃音时，发送 MESSAGE_HAVE_BACK_TONE_HALT (回铃音停止)事件给应用程序。该信号音的频率可以调用函数 BF_SetDetectBackToneFrequency 设定，缺省值为 450Hz

参　数：　ChannelNo　通道的逻辑号码。

StopTimeByMs　没有回铃音的时间，以毫秒为单位，建议取值 6000

返　回：　如果调用成功，返回 BF_OK

注　释：　如果要停止检测，需要调用函数 BF_StopDetectBackToneStop

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StopDetectBackToneStop

原　型：　unsigned short BF_StopDetectBackToneStop(unsigned short ChannelNo);

功　能：　停止识别标准的回铃音是否停止

参　数：　ChannelNo　通道的逻辑号码

返　回：　如果调用成功，返回 BF_OK

注　释：　无

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartDetectBusyTone

原　型：　unsigned short BF_StartDetectBusyTone(unsigned short ChannelNo);

功　能：　开始识别标准的忙音，当识别到忙音时，发送 MESSAGE_HAVE_BUSY_TONE(有忙音)事件给应用程序。信号音的频率可以调用函数 BF_SetDetectBusyToneFrequency（设置识别单音忙音频率函数）或 BF_SetDetectDualBusyToneFrequency（设置识别双音忙音频率函数）设定。系统缺省为单音忙音，频率缺省值为 450Hz

参　数：　ChannelNo　通道的逻辑号码

返　回：　如果调用成功，返回 BF_OK

注　释：　这个函数常常在中继模块拨号之后调用，以验证线路是否接通；或在双方通话之中调用，以检测对方是否挂机。在检测到忙音后，Digital 语音卡就停止检测标准忙音。

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StopDetectBusyTone

原　型：　unsigned short BF_StopDetectBusyTone(unsigned short ChannelNo);

功　能：　停止检测标准忙音

参　数：　ChannelNo　通道的逻辑号码

返　回：　如果调用成功，返回 BF_OK

注　释：　无

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartDetectToneData

原　型：　unsigned short BF_StartDetectToneData(unsigned short ChannelNo);

功　能：　识别标准频率的声音，定时送出识别的结果数据事件（MESSAGE_HAVE_TONE_DATA）。识别结果数据由 16 位的一个字表示，这个字的每一位的 1 或 0 分别代表有要识别的语音和没有要识别的语音。该信号音的频率可以调用函数 BF_SetDetectBusyToneFrequency 或函数 BF_SetDetectDualBusyToneFrequency 的第一个频率设定，缺省值为 450Hz

参　数：　ChannelNo　通道的逻辑号码

返　回：　如果调用成功，返回 BF_OK

注　释：　要停止识别，必须调用函数 BF_StopDetectToneData

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StopDetectToneData

原　型：　unsigned short BF_StopDetectToneData(unsigned short ChannelNo);

功　能：　停止识别信号音数据

参　数：　ChannelNo　通道的逻辑号码

返　回：　如果调用成功，返回 BF_OK
注　释：　无
适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.13 特殊信号音识别接口函数

● BF_SetSpecialSignalFrequency

原　型：　unsigned short BF_SetSpecialSignalFrequency(unsigned short ChannelNo,unsigned short Frequency);
功　能：　设置指定通道的特殊信号音，以便启动函数 BF_StartDetectSpecialSignal 开始检测
参　数：　ChannelNo　通道的逻辑号码
Frequency　特别频率，单位赫兹
返　回：　如果调用成功，返回 BF_OK
注　释：　该功能对检测传真等信号特别有用
适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartDetectSpecialSignal

原　型：　unsigned short BF_StartDetectSpecialSignal(unsigned short ChannelNo,unsigned short TimeByMilliSecond);
功　能：　该函数识别指定通道是否有特殊的信号音持续一段时间，特殊的信号音的频率有函数 BF_SetSpecialSignalFrequency 设定
参　数：　ChannelNo　通道的逻辑号码
TimeByMilliSecond　持续时间，单位为毫秒
返　回：　如果调用成功，返回 BF_OK
注　释：　在检测到该特殊信号音后，Digital 语音卡发送事件 MESSAGE_HAVE_SPECIAL_SIGNAL，并停止继续检测
适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StopDetectSpecialSignal

原　型：　unsigned short BF_StopDetectSpecialSignal(unsigned short ChannelNo);
功　能：　该函数停止检测指定通道的特别信号音
参　数：　ChannelNo　通道的逻辑号码
返　回：　如果调用成功，返回 BF_OK
注　释：　无
适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.14 话音/静音识别接口函数

● BF_SetDetectHelloVoiceThreshold

原　型：　unsigned short BF_ SetDetectHelloVoiceThreshold(unsigned short ChannelNo, unsigned short ThresholdValue);
功　能：　设置识别话音有无判断的阀值，只有当语音的能量高于该值时，才认为有语音，才能送出事件 MESSAGE_HAVE_HELLO_VOICE
参　数：　ChannelNo　通道的逻辑号码。
ThresholdValue　话音有无判断的阀值，缺省值为 0x12
返　回：　如果调用成功，返回 BF_OK
注　释：　该函数一定要在函数 BF_StartDetectHelloVoice 调用之前调用

适用性： BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartDetectHelloVoice

原　型： unsigned short BF_StartDetectHelloVoice(unsigned short ChannelNo);

功　能： 开始识别话音，当识别到有话音时，送出事件 MESSAGE_HAVE_HELLO_VOICE，并停止识别。如果需要连续识别，需要在收到事件时，再次调用该函数

参　数： ChannelNo　通道的逻辑号码

返　回： 如果调用成功，返回 BF_OK

注　释： 该函数主要用于检测对方摘机，当中继模块拨号完毕，可调用该函数，检测是否有话音，如果有，则表示对方摘机

适用性： BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StopDetectHelloVoice

原　型： unsigned short BF_StopDetectHelloVoice(unsigned short ChannelNo);

功　能： 停止识别话音

参　数： ChannelNo　通道的逻辑号码

返　回： 如果调用成功，返回 BF_OK

注　释： 无

适用性： BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartDetectSilence

原　型： unsigned short BF_StartDetectSilence(unsigned short ChannelNo,unsigned short ThresholdValue);

功　能： 开始识别静音，该函数每 12.75 毫秒产生一位数据，该位数据表示在这 12.75 毫秒内是否是静音，当该位数据值为 0 时，表示静音，否则不是静音。当 Digital 系统检测完 16 个 12.75 毫秒，即有 16 位数据时，送出事件 MESSAGE_HAVE_SILENCE_DATA，并且继续识别，直至应用程序发出停止命令。16 位数据存放在事件的 Parameter[0]字中，并且该字中高位是先检测的数据

参　数： ChannelNo　通道的逻辑号码。
ThresholdValue　语音信号的阀值，即当语音信号小于该值时，才认为是静音，建议取值为 0x12

返　回： 如果调用成功，返回 BF_OK

注　释： 该函数主要用于录音时检测静音

适用性： BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StopDetectSilence

原　型： unsigned short BF_StopDetectSilence(unsigned short ChannelNo);

功　能： 停止识别静音

参　数： ChannelNo　通道的逻辑号码

返　回： 如果调用成功，返回 BF_OK

注　释： 无

适用性： BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.15 超时设置接口函数

● BF_StartTimeOut

原　型：　unsigned short BF_StartTimeOut(unsigned short ChannelNo,unsigned short TimeBySecond);

功　能：　启动某一通道的超时计数，当相应的超时时间到达时，送出事件 MESSAGE_HAVE_TIME_OUT，然后再重新计时，当相应的超时时间到达时，再送出事件 MESSAGE_HAVE_TIME_OUT，这样周期性计时，直到停止

参　数：　ChannelNo　　通道的逻辑号码。
　　TimeBySecond　　超时时间，单位秒

返　回：　如果调用成功，返回 BF_OK

注　释：　该函数适用于包括中继模块通道、用户模块通道、录音模块通道

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StopTimeOut

原　型：　unsigned short BF_StopTimeOut(unsigned short ChannelNo);

功　能：　停止超时计时

参　数：　ChannelNo　通道的逻辑号码

返　回：　如果调用成功，返回 BF_OK

注　释：　该函数适用于包括中继模块通道、用户模块通道、录音模块通道

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_SetTimeOutValue

原　型：　unsigned short BF_SetTimeOutValue(unsigned short ChannelNo,unsigned short TimeBySecond);

功　能：　重新设置超时时间，当已经启动超时计数后，需要改变超时时间时，调用该函数

参　数：　ChannelNo　　通道的逻辑号码。
　　TimeBySecond　　新的超时时间，单位秒

返　回：　如果调用成功，返回 BF_OK

注　释：　该函数适用于包括中继模块通道、用户模块通道、录音模块通道

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.16 主叫识别和产生接口函数

● BF_StartDetectCallerID

原　型：　unsigned short BF_StartDetectCallerID(unsigned short ChannelNo);

功　能：　开始主叫识别，当识别到有主叫时，发送 MESSAGE_HAVE_FSK_CID 事件，并停止识别。如果需要继续识别，必须重新调用该函数。关于主叫信息需要调用 BF_GetFskCallerID 函数取得

参　数：　ChannelNo　　通道的逻辑号码

返　回：　如果调用成功，返回 BF_OK

注　释：　该函数适用于包括中继模块通道、用户模块通道、录音模块通道

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StopDetectCallerID

原　型：　unsigned short BF_StopDetectCallerID(unsigned short ChannelNo);

功　能：　停止主叫识别

参　数：　ChannelNo　　通道的逻辑号码

返　回：　如果调用成功，返回 BF_OK

注　释：　该函数适用于包括中继模块通道、用户模块通道、录音模块通道

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_GetRawFskCallerID

原　型：　unsigned short BF_GetRawFskCallerID(unsigned short ChannelNo,char *CallerIDString);

功　能：　获取 FSK 主叫信息，只有在收到 MESSAGE_HAVE_FSK_CID 时，调用该函数，其返回值才有意义。该函数所返回的主叫信息是未经过处理的原始信息

参　数：　ChannelNo　　通道的逻辑号码。
CallerIDString　　用户提供的存放主叫信息的缓存区，最大为 MAX_CID_NUM

返　回：　如果调用成功，返回 BF_OK

注　释：　该函数适用于包括中继模块通道、用户模块通道、录音模块通道

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_GetFskCallerID

原　型：　unsigned short BF_GetFskCallerID(unsigned short ChannelNo,char *DateTime, char *TelNo,char *Name);

功　能：　获取 FSK 主叫信息，只有在收到 MESSAGE_HAVE_FSK_CID 时，调用该函数，其返回值才有意义。该函数所返回的主叫信息是经过处理的信息，即已经分解成主叫日期时间、主叫号码和主叫名称等

参　数：　ChannelNo　通道的逻辑号码。
DateTime　主叫时间，字符串型，如果没有主叫时间，则返回空串。
TelNo　主叫号码，字符串型，如果没有主叫号码，则返回空串。
Name　主叫名称，字符串型，如果没有主叫名称，则返回空串。

返　回：　如果调用成功，返回 BF_OK

注　释：　该函数适用于包括中继模块通道、用户模块通道、录音模块通道对于 DateTime、Telno 和 Name，用户提供的缓存区长度一定要大于相应的最大可显位数，建议最好为 MAX_CID_NUM。

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartDetectDtmfCallerID

原　型：　unsigned short WINAPI BF_StartDetectDtmfCallerID(unsigned short ChannelNo);

功　能：　该函数启动 DTMF 主叫识别，必须首先调用 BF_StartDetectDTMF 函数。如果识别到 DTMF 主叫，系统有事件 MESSAGE_HAVE_CID_DTMF 产生

参　数：　ChannelNo　通道的逻辑号码

返　回：　如果调用成功，返回 BF_OK

注　释：　如果要使本函数生效，必须首先调用识别 DTMF 函数 BF_StartDetectDTMF

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StopDetectDtmfCallerID

原　型：　unsigned short WINAPI BF_StopDetectDtmfCallerID(unsigned short ChannelNo);

功　能：　该函数停止 DTMF 主叫识别

参　数：　ChannelNo　通道的逻辑号码

返　回：　如果调用成功，返回 BF_OK

注　释：　无

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_GetDtmfCallerID

原　型：unsigned short WINAPI BF_GetDtmfCallerID(unsigned short ChannelNo,char *TelNo);
功　能：如果有事件 MESSAGE_HAVE_CID_DTMF 产生,则调用该函数获取主叫内容
参　数：ChannelNo　通道的逻辑号码
TelNo　主叫信息
返　回：如果调用成功，返回 BF_OK
注　释：无
适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_SendFskCallerID

原　型：unsigned short WINAPI BF_SendFskCallerID(unsigned short ChannelNo,unsigned char *DateTime, unsigned char *TelNo);
功　能：该函数在指定的通道上发送 FSK 主叫信息，当发送完毕后，有事件 MESSAGE_SEND_FSK_END　产生
参　数：ChannelNo　通道的逻辑号码。
DateTime　主叫日期与时间，共 8 个字符，格式是 MMDDHHMM，即月月日日时时 分分，如 2 月，字符应为 02 两个字符。
TelNo　主叫电话号码，长度小于 0x40
返　回：调用成功，返回 BF_OK
注　释：该函数仅仅发送 FSK 的主叫信息，不发送振铃信息
适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.17 发送 FSK 接口函数

● BF_StarSendFSK

原　型：unsigned short BF_StartSendFsk(unsigned short ChannelNo, unsignedshort SeizureFlag,char *Buffer, unsigned short Count);
功　能：开始发送 FSK
参　数：ChannelNo　通道的逻辑号码 SeizureFlag 指示是否发送信道占用信号，当为 0 时，不　发送信道占用信号；为其他值时，发送信道占用信号。
Buffer　FSK 字符串，最大长度 MAX_FSK_NUM 个字节，字符数量由 Count 参数指定，可以包含 0x00——0xFF 在内的任意字符。
Count　FSK 字符串的字符数量，最大值为 MAX_FSK_NUM
返　回：调用成功，返回 BF_OK。发送完毕，返回发送 FSK 完毕事件
注　释：无
适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StopSendFSK

原　型：unsigned short BF_StopSendFsk(unsigned short ChannelNo);
功　能：停止发送 FSK
参　数：ChannelNo　通道的逻辑号码
返　回：如果调用成功，返回 BF_OK
注　释：无
适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.18 自动拨号接口函数

● BF_AutoDialOutWithHello

原　型：unsigned short WINAPI BF_AutoDialOutWithHello(unsigned short ChannelNo,char *DTMFString)

功　能：该函数完成摘机、检测拨号音、拨号、检测忙音、回铃音、对方摘机等功能，并返回相应的事件。该函数检测对方摘机的方法是通过语音能量

参　数：ChannelNo　通道的逻辑号码
DTMFString　电话号码

返　回：如果调用成功，返回 BF_OK

注　释：该函数所返回的事件及含义如下：

事件	含义
MESSAGE_AUTO_DIAL_NO_DIAL_TONE	没有拨号音
MESSAGE_AUTO_DIAL_NONE_LISTEN	对方没人接听
MESSAGE_AUTO_DIAL_HAVE_BUSY	忙音
MESSAGE_AUTO_DIAL_HOOK_OFF_BY_HELLO	对方摘机

适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_AutoDialOutWithPole

原　型：unsigned short WINAPI BF_AutoDialOutWithPole(unsigned short ChannelNo,char *DTMFString)

功　能：该函数完成摘机、检测拨号音、拨号、检测忙音、回铃音、对方摘机等功能，并返回相应的事件。该函数检测对方摘机的方法是通过极性反转

参　数：ChannelNo　通道的逻辑号码
DTMFString　电话号码

返　回：如果调用成功，返回 BF_OK

注　释：该函数所返回的事件及含义如下：

事件	含义
MESSAGE_AUTO_DIAL_NO_DIAL_TONE	没有拨号音
MESSAGE_AUTO_DIAL_NONE_LISTEN	对方没人接听
MESSAGE_AUTO_DIAL_HAVE_BUSY	忙音
MESSAGE_AUTO_DIAL_HOOK_OFF_BY_HELLO	对方摘机

适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_AutoDialOutWithoutDialToneWithHello

原　型：unsigned short WINAPI BF_AutoDialOutWithoutDialToneWithHello(unsigned short ChannelNo,unsigned short WaitTimeBySecond,char *DTMFString)

功　能：该函数完成摘机、等待一定时间、拨号、检测忙音、回铃音、对方摘机等功能，并返回相应的事件。该函数检测对方摘机的方法是通过语音能量

参　数：ChannelNo　通道的逻辑号码
WaitTimeBySecond　摘机之后，拨号之前等待的时间，单位为秒。
DTMFString　电话号码

返　回：如果调用成功，返回 BF_OK

注　释：该函数不检测拨号音。函数所返回的事件及含义如下：

事件	含义
MESSAGE_AUTO_DIAL_NO_DIAL_TONE	没有拨号音
MESSAGE_AUTO_DIAL_NONE_LISTEN	对方没人接听
MESSAGE_AUTO_DIAL_HAVE_BUSY	忙音
MESSAGE_AUTO_DIAL_HOOK_OFF_BY_HELLO	对方摘机

适用性： BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_AutoDialOutWithHelloNoColorRing

原　型： unsigned short WINAPI BF_AutoDialOutWithHelloNoColorRing(unsigned short ChannelNo,char *DTMFString);

功　能： 该函数完成摘机、等待一定时间、拨号、检测忙音、回铃音、对方摘机等功能，并返回相应的事件。该函数检测对方摘机的方法是通过语音能量。但是该函数会剔除彩话的影响，即对方回铃音为彩话时，不会误为语音而摘机

参　数：

ChannelNo	通道的逻辑号码
DTMFString	电话号码

返　回： 如果调用成功，返回 BF_OK

注　释： 函数所返回的事件及含义如下：

事件	含义
MESSAGE_AUTO_DIAL_NO_DIAL_TONE	没有拨号音
MESSAGE_AUTO_DIAL_NONE_LISTEN	对方没人接听
MESSAGE_AUTO_DIAL_HAVE_BUSY	忙音
MESSAGE_AUTO_DIAL_HOOK_OFF_BY_HELLO	对方摘机

适用性： BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.19 语音数据格式转换接口函数

● BF_AlawToWave

原　型： unsigned short WINAPI BF_AlawToWave(char *AlawFileName, char*WaveFileName);

功　能： 将 A-law 格式的纯语音文件转换成 16bit 线性的 Wave 格式的语音文件

参　数：

AlawFileName	A-law 语音文件名称
WaveFileName	Wave 语音文件名称

返　回： 如果调用成功，返回 BF_OK

注　释： 无

适用性： BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_WaveToAlaw

原　型： unsigned short WINAPI BF_WaveToAlaw(char *WaveFileName, char*AlawFileName);

功　能： 将 Wave 格式的语音文件转换成 16bit 线性的 A-law 格式的纯语音文件

参　数：

WaveFileName	Wave 语音文件名称
AlawFileName	A-law 语音文件名称

返　回： 如果调用成功，返回 BF_OK

注　释： 无

适用性： BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_AlawToAdpcm

原　型： unsigned short WINAPI BF_AlawToAdpcm(char *AlawFileName, char*AdpcmFileName);

功　能： 将 A-law 格式的纯语音文件转换成 32Kbit/s 的 Adpcm 格式的语音文件

参　数：

AlawFileName	A-law 语音文件名称
AdpcmFileName	32Kbit/s 的 Adpcm 语音文件名称

返　回： 如果调用成功，返回 BF_OK

注　释： 该 ADPCM 文件格式与 Dialogic ADPCM（后缀名 vox）文件格式相同

适用性： BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_AdpcmToAlaw

原　型：unsigned short WINAPI BF_AdpcmToAlaw(char *AdpcmFileName, char*AlawFileName);
功　能：将 32Kbit/s 的 Adpcm 格式的语音文件转换成 A-law 格式的纯语音文件
参　数：AdpcmFileName　32Kbit/s 的 Adpcm 语音文件名称
　　AlawFileName　A-law 语音文件名称
返　回：如果调用成功，返回 BF_OK
注　释：该 ADPCM 文件格式与 Dialogic ADPCM（后缀名 vox）文件格式相同
适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_G711ToG726

原　型：unsigned short WINAPI BF_G711ToG726(char *G711FileName,char *G726FileName, unsigned short G726VoiceType);
功　能：该函数将格式为 A 率的 G.711 格式的文件转换为 ADPCM 的 G726 格式的文件, G726 文件格式由参数 G726VoiceType 指定
参　数：

ChannelNo	通道的逻辑号码。
G711FileName	G711、A-Law 格式文件名称。
G726FileName	G726、ADPCM 格式文件名称。
G726VoiceType	G726 文件格式类型，取值如下：
VOICE_TYPE_G726_32K	G.726 ADPCM 32Kbit/s。
VOICE_TYPE_G726_16K	G.726 ADPCM 16Kbit/s

返　回：如果调用成功，返回 BF_OK
注　释：无
适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_G726ToG711

原　型：unsigned short WINAPI BF_G726ToG711(char *G726FileName,unsigned short G726VoiceType, char *G711FileName);
功　能：该函数将格式为 ADPCM 的 G726 格式的文件转换为 A 率的 G.711 格式的文件, G726 文件格式由参数 G726VoiceType 指定
参　数：

ChannelNo	通道的逻辑号码。
G711FileName	G711、A-Law 格式文件名称。
G726FileName	G726、ADPCM 格式文件名称。
G726VoiceType	G726 文件格式类型，取值如下：
VOICE_TYPE_G726_32K	G.726 ADPCM 32Kbit/s。
VOICE_TYPE_G726_16K	G.726 ADPCM 16Kbit/s

返　回：如果调用成功，返回 BF_OK
注　释：无
适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.20 停止通道所有操作接口函数

● BF_StopChannelOperate

原　型：unsigned short BF_StopChannelOperate(unsigned short ChannelNo);
功　能：停止通道的一切操作，仅指语音通道，不包括资源通道，使通道恢复打开时的状态
参　数：ChannelNo　通道的逻辑号码
返　回：如果调用成功，返回 BF_OK
注　释：无
适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.21 会议接口函数

● BF_ConferenceAddMember

原　型：unsigned short BF_ConferenceAddMember(unsigned short ChannelNo, unsigned short Mode, unsigned short Group);

功　能：将指定通道添加到某组会议中

参　数：ChannelNo　通道的逻辑号码，该号码可以大于系统的实际通道数量，如果大于实际通道表示使用虚拟通道，虚拟通道加入的是 IP 会议，即将与该虚拟通道绑定的 IP 资源加入会议。加入 IP 会议的成员的模式，仅仅是讲话模式和只听模式，不能是录音模式。

Mode　指示添加模式，取值如下：

1——讲话模式

2——只听模式

3——录音模式

Group　会议的组别，最大为 MAX_CONFERENCE_GROUP_NUM

返　回：调用成功，返回 BF_OK

注　释：每增加一个讲话/录音模式，占用一个会议资源；在同一个会议组里，所有只听模式共用一个会议资源，所有录音模式共用一个会议资源。

适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8

● BF_ConferenceDelMember

原　型：unsigned short BF_ConferenceDelMember(unsigned short ChannelNo,unsigned short Mode, unsigned short Group);

功　能：将指定通道从某组会议中删除

参　数：ChannelNo　通道的逻辑号码

Mode　指示添加时的模式，取值如下：

1——讲话模式

2——只听模式

3——录音模式

Group　会议的组别，最大为 MAX_CONFERENCE_GROUP_NUM

返　回：调用成功，返回 BF_OK

注　释：无

适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8

● BF_ConferenceAddMemberWithGain

原　型：unsigned short WINAPI BF_ConferenceAddMemberWithGain(unsigned short ChannelNo, unsigned short Group, short Gain);

功　能：该函数的功能与 BF_ConferenceAddMember 中的加入讲话模式的会议功能相同,即该函数固定增加讲话模式的会议成员，但该函数可以设置加入会议者说话时的音量增益，即增加了 Gain 参数

参　数：ChannelNo　通道的逻辑号码。，该号码可以大于系统的实际通道数量，如果大于实际通道表示使用虚拟通道，虚拟通道加入的是 IP 会议，即将与该虚拟通道绑定的 IP 资源加入会议。

Group　会议的组别，最大为 MAX_CONFERENCE_GROUP_NUM

Gain　话音增益，建议取值 0；取值范围在-16 至 15 之间，取值越大，声音越大；取值越小，声音越小。

返　回：调用成功，返回 BF_OK

注　释：每增加一个讲话模式，占用一个会议资源

适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8

3.2.22 TTS(Text To Speech)接口函数

Digital 系统含有一套免费的简单的 TTS 系统，能将文本转换成语音。如果用户的要求严格，Digital 语音卡可以提供第三方的语音合成系统，以满足客户的需求。

● BF_InitTTS

原　型：　unsigned short BF_InitTTS(void);
功　能：　初始化 TTS 系统，该函数仅仅调用一次，只有返回成功时，TTS 系统才能正常工作
参　数：　无
返　回：　如果调用成功，返回 BF_OK
注　释：　无
适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_CloseTTS

原　型：　unsigned short BF_CloseTTS(void);
功　能：　关闭 TTS 系统，在不使用 TTS 系统时，调用该函数，以释放系统资源
参　数：　无
返　回：　如果调用成功，返回 BF_OK
注　释：　无
适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_TTSFileToFile

原　型：　unsigned short BF_TTSFileToFile(char *TextFileName, char *VoiceFileName);
功　能：　将文本文件转换成语音文件
参　数：　TextFileName　　文本文件名称
　　VoiceFileName　　语音文件名称
返　回：　如果调用成功，返回 BF_OK
注　释：　无
适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_TTSBufferToBuffer

原　型：　unsigned short BF_TTSBufferToBuffer(char *TextBuffer,unsigned long TextCount, char *VoiceBuffer,unsigned long *VoiceCount);
功　能：　将文本缓存区转换成语音缓存区
参　数：　TextBuffer　　文本缓存区指针
　　TextCount　　文本的字节数量
　　VoiceBuffer　　语音缓存区指针
　　VoiceCount　　语音缓存区数据数量，单位字节
返　回：　如果调用成功，返回 BF_OK
注　释：　无
适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_TTSBufferToFile

原　型：　unsigned short BF_TTSBufferToFile(char *TextBuffer,unsigned long TextCount, char *VoiceFileName);
功　能：　将文本缓存区转换成语音文件

参　数：　TextBuffer　　文本缓存区指针
　　TextCount　　文本的字节数量
　　VoiceFileName　　语音文件名称
返　回：　如果调用成功，返回 BF_OK
注　释：　无
适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.23 语音数据流处理函数

为支持实时的语音处理需要，如网络监听、VoIP 等，Digital 系列语音卡特提供两个语音数据流处理函数。

● BF_ReadVoicePacket

原　型：　unsigned short WINAPI BF_ReadVoicePacket(unsigned short ChannelNo, char *Buffer);
功　能：　该函数读取特定通道的语音数据流，每次读取 512 字节，固定为 A-law 数据
参　数：　ChannelNo　　通道的逻辑号码。
　　Buffer　　如果读取数据成功，则存放读取回来的语音数据，长度为 512 字节。
返　回：　如果调用成功，返回 BF_OK
注　释：　无
适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-PCI-R16、BF-PCI-R8

● BF_WriteVoicePacket

原　型：　unsigned short WINAPI BF_WriteVoicePacket(unsigned short ChannelNo, char *Buffer);
功　能：　该函数写入语音数据流到特定通道，每次写入 512 字节，固定为 A-law 数据。
参　数：　ChannelNo　　通道的逻辑号码。
　　Buffer　　欲写入的语音数据，长度为 512 字节
返　回：　如果调用成功，返回 BF_OK
注　释：　只有返回 BF_OK，才表示 Buffer 的数据成功写入，否则应该再次写入该数据。一般情况下，64 毫秒应写入一个数据包
适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-PCI-R16、BF-PCI-R8

3.2.24 特殊语音格式的录音放音处理函数

Digital 系列语音卡支持 G.711 和 G.726 语音格式，用户可以根据需要，选择合适的放音和录音的语音数据格式。

● BF_SetVoiceType

原　型：　unsigned short WINAPI BF_SetVoiceType(unsigned short ChannelNo, unsigned short VoiceType);
功　能：　该函数设置放音和录音的语音格式，该函数仅仅影响函数 BF_StartPlayFileByVoiceType、BF_StartRecordFileByVoiceType、BF_StartPlayFileByFileType 和 BF_StartRecordFileByFileType。
参　数：　ChannelNo　通道的逻辑号码。
　　VoiceType　语音格式，取值如下：

VOICE_TYPE_A_LAW	G.711 A 率 64Kbit/s。
VOICE_TYPE_G726_32K	G.726 ADPCM 32Kbit/s。
VOICE_TYPE_G726_16K	G.726 ADPCM 16Kbit/s。

VOICE_TYPE_DIALOGIC_ADPCM	Dialogic ADPCM 32Kbit/s。
VOICE_TYPE_LINEAR_8BIT	8 位线性，无符号 64 Kbit/s。
VOICE_TYPE_LINEAR_16BIT	16 位线性，无符号 128 Kbit/s。

返　回：　如果调用成功，返回 BF_OK

注　释：　如果调用 BF_StartPlayFileByVoiceType 、BF_StartRecordFileByVoiceType 、BF_StartPlayFileByFileType 和 BF_StartRecordFileByFileType 之外的函数进行录音或放音，则会改变录音或放音的语音格式为 G.711 A 率 64Kbit/s。

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_SetPlayVoiceType

原　型：　unsigned short WINAPI BF_SetPlayVoiceType(unsigned short ChannelNo,unsigned short VoiceType);

功　能：　该函数仅仅设置放音的语音格式，该函数仅仅影响函数 BF_StartPlayFileByVoiceType 和 BF_StartPlayFileByFileType

参　数：　ChannelNo　通道的逻辑号码。

VoiceType　语音格式，取值如下：

VOICE_TYPE_A_LAW	G.711 A 率 64Kbit/s。
VOICE_TYPE_G726_32K	G.726 ADPCM 32Kbit/s。
VOICE_TYPE_G726_16K	G.726 ADPCM 16Kbit/s。
VOICE_TYPE_DIALOGIC_ADPCM	Dialogic ADPCM 32Kbit/s。
VOICE_TYPE_LINEAR_8BIT	8 位线性，无符号 64 Kbit/s。
VOICE_TYPE_LINEAR_16BIT	16 位线性，无符号 128 Kbit/s。

返　回：　如果调用成功，返回 BF_OK

注　释：　如果调用 BF_StartPlayFileByVoiceType 和 BF_StartPlayFileByFileType 之外的函数进行放音，则会改变放音的语音格式为 G.711 A 率 64Kbit/s。

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_SetRecordVoiceType

原　型：　unsigned short WINAPI BF_SetRecordVoiceType(unsigned short ChannelNo, unsigned short VoiceType);

功　能：　该函数仅仅设置录音的语音格式，该函数仅仅影响函数 BF_StartRecordFileByVoiceType 和 BF_StartRecordFileByFileType

参　数：　ChannelNo　通道的逻辑号码。

VoiceType　语音格式，取值如下：

VOICE_TYPE_A_LAW	G.711 A 率 64Kbit/s。
VOICE_TYPE_G726_32K	G.726 ADPCM 32Kbit/s。
VOICE_TYPE_G726_16K	G.726 ADPCM 16Kbit/s。
VOICE_TYPE_DIALOGIC_ADPCM	Dialogic ADPCM 32Kbit/s。
VOICE_TYPE_LINEAR_8BIT	8 位线性，无符号 64 Kbit/s。
VOICE_TYPE_LINEAR_16BIT	16 位线性，无符号 128 Kbit/s。

返　回：　如果调用成功，返回 BF_OK

注　释：　如果调用 BF_StartRecordFileByVoiceType 和 BF_StartRecordFileByFileType 之外的函数进行录音，则会改变录音的语音格式为 G.711 A 率 64Kbit/s

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartPlayFileByVoiceType

原　型：　unsigned short WINAPI BF_StartPlayFileByVoiceType(unsigned short ChannelNo, char *FileName,unsigned long Offset,unsigned long Length,char StopChar);

功　能：　该函数按照函数 BF_SetVoiceType 或函数 BF_SetPlayVoiceType 设置的语音格式进行放音

参　数：　hannelNo　通道的逻辑号码。

	FileName	如果放音数据在文件中，则指定文件名称，含有文件存放的路径，最大长度为 MAX_FILE_NAME_NUM-1
	Ofsset	指定放音文件的字节偏移长度
	Length	指定放音文件的字节长度，如果设为 0xFFFFFFFF，则表示一直到文件结尾
	StopChar	停止放音的 DTMF，当收到相应的 DTMF 按键时，停止放音，有效取值为：
		0x00 表示收到任何 DTMF 按键都不停止
		0x01 表示收到任一 DTMF 按键就停止
		0—9,*,#,A,B,C,D 表示收到某个 DTMF 按键就停止
		其他值 没有意义

返　回：　如果调用成功，返回 BF_OK

注　释：　如果要使 StopChar 生效，必须首先调用识别 DTMF 函数 BF_StartDetectDTMF

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartRecordFileByVoiceType

原　型：　unsigned short WINAPI BF_StartRecordFileByVoiceType(unsigned short ChannelNo, char *FileName,unsigned long Offset,unsigned long Length,char StopChar);

功　能：　该函数按照函数 BF_SetVoiceType 或 BF_SetRecordVoiceType 函数设置的语

参　数：

ChannelNo	通道的逻辑号码。
FileName	录音文件名称，含有文件存放的路径，最大长度为 MAX_FILE_NAME_NUM-1。
Offset	如果录音文件存在，存放语音的起始位置，如果录音文件不存在，必须设置为 0。
Length	需要录音数据的字节长度，如果设为 0xFFFFFFFF，则一直录音，直到发送录音停止命令，或有指定的 DTMF 收到。
StopChar	停止录音的 DTMF，当收到相应的 DTMF 按键时，停止录音，有效取值为：
	0x00 表示收到任何 DTMF 按键都不停止
	0x01 表示收到任一 DTMF 按键就停止
	0—9,*,#,A,B,C,D 表示收到某个 DTMF 按键就停止
	其他值 没有意义

返　回：　如果调用成功，返回 BF_OK

注　释：　无

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.25 特殊文件格式的录音放音处理函数

Digital 系列语音卡支持纯语音数据文件格式和.WAV 文件格式两种格式的语音文件，用户可以根据需要，选择合适的放音和录音的文件格式。

● BF_SetPlayVoiceFileType

原　型：　BF_SetPlayVoiceFileType(unsigned short ChannelNo, unsigned short FileType);

功　能：　该函数设置放音文件的文件类型，该函数仅仅影响函数

参　数：

ChannelNo	通道的逻辑号码。
FileType	语音格式，取值如下：
VOICE_FILE_TYPE_NORMAL	纯语音数据的文件格式。
VOICE_FILE_TYPE_WAVE	Microsoft .WAV 的文件格式

返　回：　如果调用成功，返回 BF_OK

注　释：　无

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_SetRecordVoiceFileType

原　型：BF_SetRecordVoiceFileType(unsigned short ChannelNo, unsigned short FileType);

功　能：该函数设置录音文件的文件类型，该函数仅仅影响函数 BF_StartRecordFileByFileType

参　数：hannelNo 通道的逻辑号码。

FileType 语音格式，取值如下：

VOICE_FILE_TYPE_NORMAL	纯语音数据的文件格式。
VOICE_FILE_TYPE_WAVE	Microsoft .WAV 的文件格式

返　回：如果调用成功，返回 BF_OK

注　释：无

适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartPlayFileByFileType

原　型：unsigned short WINAPI BF_StartPlayFileByFileType(unsigned shortChannelNo, char *FileName,unsigned long Offset,unsigned long Length,char StopChar);

功　能：该函数按照函数 BF_SetVoiceType 或函数 BF_SetPlayVoiceType 设置的语音格式、按照函数 BF_SetPlayVoiceFileType 设置的文件类型进行放音

参　数：ChannelNo　通道的逻辑号码。

FileName 如果放音数据在文件中，则指定文件名称，含有文件存放的路径，最大长度为 MAX_FILE_NAME_NUM - 1

Offset　该参数没有意义，必须设置为 0。

Length　指定放音文件的字节长度，如果设为 0xFFFFFFFF，则表示一直到文件结尾

StopChar　停止放音的 DTMF，当收到相应的 DTMF 按键时，停止放音，有效取值为：

0x00	表示收到任何 DTMF 按键都不停止
0x01	表示收到任一 DTMF 按键就停止
0—9,*,#,A,B,C,D	表示收到某个 DTMF 按键就停止
其他值	没有意义

返　回：如果调用成功，返回 BF_OK

注　释：如果要使 StopChar 生效，必须首先调用识别 DTMF 函数 BF_StartDetectDTMF

适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartRecordFileByFileType

原　型：unsigned short WINAPI BF_StartRecordFileByFileType(unsigned short ChannelNo, char *FileName,unsigned long Offset,unsigned long Length,char StopChar);

功　能：该函数按照函数 BF_SetVoiceType 或 BF_SetRecordVoiceType 函数设置的语音格式、按照函数 BF_SetRecordVoiceFileType 设置的文件类型进行录音

参　数：ChannelNo　通道的逻辑号码。

FileName　录音文件名称，含有文件存放的路径，最大长度为 MAX_FILE_NAME_NUM-1。

Offset　该参数没有意义，必须设置为 0。

Length　需要录音数据的字节长度，如果设为 0xFFFFFFFF，则一直录音，直到发送录 音停止命令，或有指定的 DTMF 收到。

StopChar　停止录音的 DTMF，当收到相应的 DTMF 按键时，停止录音，有效取值为：

0x00	表示收到任何 DTMF 按键都不停止
0x01	表示收到任一 DTMF 按键就停止
0—9,*,#,A,B,C,D	表示收到某个 DTMF 按键就停止

返　回：如果调用成功，返回 BF_OK

注　释：如果要使 StopChar 生效，必须首先调用识别 DTMF 函数 BF_StartDetectDTMF

适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartPlayAdpcmFile

原　型：unsigned short WINAPI BF_StartPlayAdpcmFile(unsigned shortChannelNo,char *FileName,unsigned long Offset,unsigned longLength,char StopChar);

功　能：该函数播放格式为 Dialogic ADPCM(文件后缀为 vox)的语音文件

参　数：ChannelNo　通道的逻辑号码。

FileName　播放语音文件名称，含有文件存放的路径，最大长度为 MAX_FILE_NAME_NUM-1。

Offset　该参数没有意义，必须设置为 0。

Length　需要播放数据的字节长度，如果设为 0xFFFFFFFF，则一直播放，直到发送录音停止命令，或有指定的 DTMF 收到。

StopChar　停止播放的 DTMF，当收到相应的 DTMF 按键时，停止播放，有效取值为：

0x00	表示收到任何 DTMF 按键都不停止
0x01	表示收到任一 DTMF 按键就停止
0—9,*,#,A,B,C,D	表示收到某个 DTMF 按键就停止
其他值	没有意义

返　回：如果调用成功，返回 BF_OK

注　释：如果要使 StopChar 生效，必须首先调用识别 DTMF 函数 BF_StartDetectDTMF

适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartRecordAdpcmFile

原　型：unsigned short WINAPI BF_StartRecordAdpcmFile(unsigned short;hannelNo, char *FileName,unsigned long Offset,unsigned longLength,char StopChar);

功　能：该函数录制格式为 Dialogic ADPCM(文件后缀为 vox)的语音文件

参　数：ChannelNo　通道的逻辑号码。

FileName　录音文件名称，含有文件存放的路径，最大长度为 MAX_FILE_NAME_NUM-1。

Offset　该参数没有意义，必须设置为 0。

Length　需要录音数据的字节长度，如果设为 0xFFFFFFFF，则一直录音，直到发送录音停止命令，或有指定的 DTMF 收到。

StopChar　停止录音的 DTMF，当收到相应的 DTMF 按键时，停止录音，有效取值为：

0x00	表示收到任何 DTMF 按键都不停止
0x01	表示收到任一 DTMF 按键就停止
0—9,*,#,A,B,C,D	表示收到某个 DTMF 按键就停止
其他值	没有意义

返　回：如果调用成功，返回 BF_OK

注　释：无

适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StartPlayColorBackground

原　型：unsigned short WINAPI BF_StartPlayColorBackground(unsigned shortChannelNo, char *FileName,unsigned long Offset,unsigned long Length,char StopChar,short VoiceVolume)

功　能：该函数将指定的语音文件与号码为 ChannelNo 的模拟通道语音混合后输出，实现播放背景音的功能，即彩话功能

参　数：ChannelNo　通道的逻辑号码。

FileName　录音文件名称，该文件只能是 A-law 格式。

Offset　需要放音的起始位置。

Length　需要放音数据的字节长度，如果设为 0xFFFFFFFF，则一直放音，直到发送放音停止命令，或有指定的 DTMF 收到。

StopChar 停止录音的 DTMF，当收到相应的 DTMF 按键时，停止录音，有效取值为：
0x00 表示收到任何 DTMF 按键都不停止
0x01 表示收到任一 DTMF 按键就停止
0—9,*,#,A,B,C,D 表示收到某个 DTMF 按键就停止其他值没有意义
VoiceVolume 语音增益，仅仅对播放文件的语音起作用，取值范围在-5 至+5 之间，0 为原声，没有任何增益；负数为增加的增益倍数；正数为减小的增益倍数。

返　回：如果调用成功，返回 BF_OK
注　释：即使收到放音结束事件，也必须调用函数 BF_ StopPlayColorBackground
适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_StopPlayColorBackground

原　型：unsigned short BF_ StopPlayColorBackground (unsigned short ChannelNo);
功　能：停止播放背景音操作。
参　数：ChannelNo　通道的逻辑号码
返　回：如果调用成功，返回 BF_OK
注　释：无
适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.26 混音处理函数

在正常情况下，内线和外线使用函数 BF_LinkChannel 连通后，如果对其中的一个通道录音，虽然可以录两个通道的声音，但是会有其中一个通道的声音很小。为了解决上述问题，特提供下面两个函数，对连通的通道进行混音，以使两个通道录音的声音大小平衡。

● BF_StartMixVoice

原　型：unsigned short WINAPI BF_StartMixVoice(unsigned short FirstChannelNo,unsigned short SecondChannelNo);
功　能：该函数将两个通道的声音混合在一起，以便录音。也就是说，在调用该函数后，对这两个通道的任意一个通道录音，录的音都是这两个通道混合的声音，即两个通道的声音。
参　数：FirstChannelNo　通道的逻辑号码。
SecondChannelNo　通道的逻辑号码。
返　回：如果调用成功，返回 BF_OK
注　释：该函数主要用于内线和外线连通后，对双方录音
适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8

● BF_StopMixVoice

原　型：unsigned short WINAPI BF_StopMixVoice(unsigned short FirstChannelNo,unsigned short SecondChannelNo)
功　能：该函数停止两个通道的混音
参　数：FirstChannelNo　通道的逻辑号码。
SecondChannelNo　通道的逻辑号码。
返　回：如果调用成功，返回 BF_OK
注　释：无
适用性：BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8

● BF_MixVoiceFile

原　型：unsigned short WINAPI BF_MixVoiceFile(char *FirstFileName, char *SecondFileName, char

*TargetFileName, unsigned short MixFlag);

功　能：　该函数将两个 A-Law(A 律)格式的语音文件混合在一起

参　数：　FirstFileName　　第一个源 A-Law 语音文件名称。
SecondFileName　第二个源 A-Law 语音文件名称。
TargetFileName　混合后的 A-Law 语音文件名称
MixFlag　　是否循环混合标志，取值如下：
0 —不循环混合，即两个长度不同的文件，从两个文件头开始混合，如果某个文件结束，则将另一个文件的余下内容直接写入目标文件。
1 —循环混合，即两个长度不同的文件，从两个文件头开始混合，如果某个文件结束，则回到该文件头读取数据，继续与另一个文件混合，直至较长文件结束为止

返　回：　如果调用成功，返回 BF_OK

注　释：　无

适用性：　BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

3.2.27 点亮熄灭指示灯接口函数

● BF_ControlLed

原　型：　unsigned short WINAPI BF_ControlLed(unsigned short LedNo, unsigned short LightFlag);

功　能：　该函数点亮或熄灭板上的指示灯,板上共有五个指示等,详细叙述如下:
左侧指示灯：单一红色的指示灯,是电源指示灯,板卡供电即亮;
中间指示灯：是一个红绿双色指示灯，其中红色等定义为 0 号灯，绿色等定义为 1 号灯。
右侧指示灯：是一个红绿双色指示灯，其中红色等定义为 2 号灯，绿色等定义为 3 号灯

参　数：　无

返　回：　如果调用成功，返回 BF_OK

注　释：　LedNo　　灯的序号。
LightFlag　　点亮或熄灭指示灯，0 为熄灭，1 为点亮。

适用性：　BF-USB-V2

3.2.28 读写用户信息接口函数

USB 系列卡为用户提供了 64 字节的硬件存储区，用于存储用户的硬件信息，写入该区域的数据不会因设备掉电而丢失，可以用于用户应用程序的加密。该区域的读写由下面程序完成。

● BF_WriteUserDataToUSB

原　型：　unsigned short WINAPI BF_WriteUserDataToUSB(unsigned char Address, unsigned char Length, unsigned char *DataBuffer);

功　能：　该函数已经废弃，请停止使用。

参　数：

返　回：

注　释：　该函数已经废弃，请停止使用。

适用性：　BF-USB-V2

● BF_ReadUserDataFromUSB

原　型：　unsigned short WINAPI BF_ ReadUserDataFromUSB(unsigned char Address, unsigned char Length, unsigned char *DataBuffer);

功　能：　该函数已经废弃，请停止使用。

参　数：

返　回：

注　释：　该函数已经废弃，请停止使用。

适用性：　BF-USB-V2

第四部分 Digital 语音卡传真处理部分接口函数说明

Digital 语音卡包含资源通道，以提供传真、VoIP 和语音压缩等功能使用。

Digital 语音卡所有的资源通道在打开操作之后，必须与相应的语音通道绑定，然后在语音通道上，操作资源的功能。下面具体介绍资源的操作接口函数。

第一章 Digital 语音卡传真部分编程接口函数索引表（按功能分类）

获取系统参数接口函数

```
unsigned short WINAPI BF_GetTotalResource(void);
unsigned short WINAPI BF_GetResourceType(unsigned short ResourceNo);
```

打开/关闭资源接口函数

```
unsigned short WINAPI BF_OpenResource(unsigned short ResourceNo,unsigned short
        ResourceType);
unsigned short WINAPI BF_CloseResource(unsigned short FaxChannelNo);
```

绑定资源接口函数

```
unsigned short WINAPI BF_BindResource(unsigned short ChannelNo, unsigned short
        ResourceNo);
unsigned short WINAPI BF_FreeResource(unsigned short ChannelNo, unsigned short
        ResourceNo);
```

传真资源操作接口函数

```
unsigned short WINAPI BF_InitFaxFunction(void);
unsigned short WINAPI BF_CloseFaxFunction(void);
unsigned short WINAPI BF_SetFaxLocalTelNo(unsigned short ChannelNo,unsigned char
        *LocalTelNo);
unsigned short WINAPI BF_GetFaxRemoteTelNo(unsigned short ChannelNo,unsigned char
        *RemoteTelNo);
unsigned short WINAPI BF_SetFaxResolution(unsigned short ChannelNo,unsigned short
        FaxResolution);
unsigned short WINAPI BF_GetFaxResolution(unsigned short ChannelNo,unsigned short
        *FaxResolution);
unsigned short WINAPI BF_SetFaxHeader(unsigned short ChannelNo,char *FaxFrom,char
        *FaxTo,char *FaxNo,char *FaxSubject,char *DateTime);
unsigned short WINAPI BF_SetFaxSendHeaderFlag(unsigned short ChannelNo, unsigned
        short FaxSendHeaderFlag);
unsigned short WINAPI BF_SetFaxMaxRate(unsigned short ChannelNo, unsigned short
        FaxMaxRate);
unsigned short WINAPI BF_GetFaxInformation(unsigned short ChannelNo,
```

```
        PBF_FAX_INFORMATION FaxInformation);
unsigned short WINAPI BF_GetFaxPageNo(unsigned short ChannelNo);
unsigned short WINAPI BF_SendFaxTxtBuffer(unsigned short ChannelNo, char
        *TxtBuffer, unsigned short BufferCount);
unsigned short WINAPI BF_SendFaxTxtFile(unsigned short ChannelNo,char *FileName);
unsigned short WINAPI BF_SendBitFaxFile(unsigned short ChannelNo,char *FileName);
unsigned short WINAPI BF_SendMultiBitFaxFile(unsigned short ChannelNo,char
        *Directory,char *FileName);
unsigned short WINAPI BF_SendBitFaxFileWithPageNo(unsigned short ChannelNo,
        unsigned short StartPageNo, unsigned short EndPageNo, char *FileName);
unsigned short WINAPI BF_SendTiffFaxFile(unsigned short ChannelNo, char
        *FileName);
unsigned short WINAPI BF_SendTiffFaxFileWithPageNo(unsigned short ChannelNo,
        unsigned short StartPageNo, unsigned short EndPageNo, char *FileName);
unsigned short WINAPI BF_ReceiveBitFaxFile(unsigned short ChannelNo, char
        *FileName);
unsigned short WINAPI BF_ReceiveTiffFaxFile(unsigned short ChannelNo, char
        *FileName);
unsigned short WINAPI BF_SendMultiTiffFaxFile(unsigned short ChannelNo,
        PSEND_MULTI_FAX_INFO FileInfo,unsigned short FileCount);
unsigned short WINAPI BF_GetFaxLength(unsigned short ChannelNo, unsigned long
        *FaxLength);
unsigned short WINAPI BF_StopFaxOperate(unsigned short ChannelNo);
```

第二章 Digital 语音卡传真部分编程接口函数详细说明

4.2.1 获取系统参数接口函数

● BF_GetTotalResource

原　型：　unsigned short WINAPI BF_GetTotalResource(void);
功　能：　获取系统的资源总的数量
参　数：　无
返　回：　如果调用成功，返回系统资源总的数量
注　释：　无
适用性：　BF-PCI-V8F4、BF-PCI-V8F8

● BF_GetResourceType

原　型：　unsigned short WINAPI BF_GetResourceType(unsigned short ResourceNo);
功　能：　获取资源的类型
参　数：　ResourceNo　　资源的逻辑通道号码
返　回：　如果调用成功，返回资源的类型
注　释：　无
适用性：　BF-PCI-V8F4、BF-PCI-V8F8

4.2.2 打开/关闭资源接口函数

● BF_OpenResource

原　型：unsigned short WINAPI BF_OpenResource(unsigned short ResourceNo, unsigned short ResourceType);

功　能：打开资源通道。在使用资源以前，必须按需求打开资源通道

参　数：ResourceNo　　资源的逻辑通道号码

ResourceType　　打开的类型，某个资源通道可能适于多种类型，如即适于传真，又适于 VoIP，但只能按一种类型打开，该参数的取值如下：

RESOURCE_TYPE_FAX	0x0001	传真资源
RESOURCE_TYPE_VOIP	0x0002	VoIP 资源
RESOURCE_TYPE_CODEC	0x0004	编码/解码资源

返　回：如果调用成功，返回 BF_OK

注　释：无

适用性：BF-PCI-V8F4、BF-PCI-V8F8

● BF_CloseResource

原　型：unsigned short WINAPI BF_CloseResource(unsigned short FaxChannelNo);

功　能：关闭打开的资源

参　数：ResourceNo　　资源的逻辑通道号码

返　回：如果调用成功，返回 BF_OK

注　释：无

适用性：BF-PCI-V8F4、BF-PCI-V8F8

4.2.3 绑定资源接口函数

● BF_BindResource

原　型：unsigned short WINAPI BF_BindResource(unsigned short ChannelNo, unsigned short ResourceNo);

功　能：绑定资源。任何打开的资源，必须与相应的语音通道绑定之后才能体现其功能，在绑定之后，可以直接在语音通道上，操作被绑定的资源。

参　数：ChannelNo　　语音通道的逻辑通道号码

ResourceNo　　资源通道的逻辑通道号码

返　回：如果调用成功，返回 BF_OK

注　释：在调用 BF_BindResource 之后，没有调用 BF_FreeResource 释放资源，相应资源将无法被再次绑定。

适用性：BF-PCI-V8F4、BF-PCI-V8F8

● BF_FreeResource

原　型：unsigned short WINAPI BF_FreeResource(unsigned short ChannelNo, unsigned short ResourceNo);

功　能：释放被绑定的资源。当语音通道不需要资源操作时，必须释放被绑定的资源，BF_StopChannelOperate 函数并不能释放被绑定的资源。

参　数：ChannelNo　　语音通道的逻辑通道号码

ResourceNo　　资源通道的逻辑通道号码

返　回：如果调用成功，返回 BF_OK

注　释：BF_StopChannelOperate 函数并不能释放被绑定的资源。

适用性： BF-PCI-V8F4、BF-PCI-V8F8

4.2.4 传真资源操作接口函数

Digital 系列语音卡的某些板卡带有传真资源，这些传真资源可以被插入同一台机器上的指定 Digital 系列语音卡的语音通道使用，使用方法如下：

(1) 调用函数 BF_InitFaxFunction 来初始化传真功能

(2) 调用函数 BF_OpenResource 以 RESOURCE_TYPE_FAX 打开资源通道

(3) 调用函数 BF_BindResource 绑定资源通道与语音通道

(4) 调用函数 BF_SetFaxLocalTelNo 设置相应语音通道的本地传真号码

(5) 调用相关函数 BF_SendFaxTxtBuffer、BF_SendFaxTxtFile、BF_SendBitFaxFile、BF_SendTiffFaxFile、BF_SendMultiTiffFaxFile、BF_ReceiveBitFaxFile 和 BF_ReceiveTiffFaxFile 进行发送或接收传真

(6) 如果收到事件 MESSAGE_FAX_HAVE_REMOTE_TEL_NO，可以调用函数 BF_GetFaxRemoteTelNo 取得对方传真号码

(7) 直到收到相应事件，如 MESSAGE_FAX_SEND_OK、MESSAGE_FAX_SEND_ERROR、MESSAGE_FAX_RECEIVE_OK、MESSAGE_FAX_RECEIVE_ERROR 再进行下一步操作

(8) 调用函数 BF_FreeResource 释放资源

(9) 如果还有传真操作，从步骤 3 开始执行

(10) 当系统退出时，调用函数 BF_CloseFaxFunction 释放系统资源。

● BF_InitFaxFunction

原　型： unsigned short BF_InitFaxFunction(void);
功　能： 初始化传真系统，该函数仅仅调用一次，只有返回成功时，传真系统才能正常工作。
参　数： 无
返　回： 如果调用成功，返回 BF_OK
注　释： 无
适用性： BF-PCI-V8F4、BF-PCI-V8F8

● BF_CloseFaxFunction

原　型： unsigned short BF_CloseFaxFunction(void);
功　能： 关闭传真系统
参　数： 无
返　回： 如果调用成功，返回 BF_OK
注　释： 无
适用性： BF-PCI-V8F4、BF-PCI-V8F8

● BF_SetFaxLocalTelNo

原　型： unsigned short BF_SetFaxLocalTelNo(unsigned short ChannelNo, unsigned char *LocalTelNo);
功　能： 设置指定通道电话号码，该号码将发送给对方传真机。
参　数： ChannelNo　　语音通道号码，从 0 开始
LocalTelNo　　电话号码
返　回： 如果调用成功，返回 BF_OK
注　释： 该函数一定要在调用 BF_BindResource 函数之后调用，才能有效。

适用性： BF-PCI-V8、BF-PCI-V8F4、BF-PCI-V8F8、BF-USB-V2、BF-PCI-R16、BF-PCI-R8

● BF_GetFaxRemoteTelNo

原　型： unsigned short BF_GetFaxRemoteTelNo(unsigned short ChannelNo, unsigned char *RemoteTelNo)

功　能： 获取对方传真机电话号码

参　数： ChannelNo　　语音通道号码，从 0 开始
RemoteTelNo　　对方传真电话号码

返　回： 如果调用成功，返回 BF_OK

注　释： 该函数只有在收到 MESSAGE_FAX_HAVE_REMOTE_TEL_NO 事件后调用，才有意义

适用性： BF-PCI-V8F4、BF-PCI-V8F8

● BF_SetFaxResolution

原　型： unsigned short WINAPI BF_SetFaxResolution(unsigned short ChannelNo, unsigned short FaxResolution);

功　能： 该函数设置传真的分辨率，有高低两种分辨率：
当设置为高分辨率时，系统优先选择高分辨率进行握手。
如果对方传真机使用低分辨率，则本端自动使用低分辨率；
如果对方传真机使用高分辨率，则本端也使用高分辨率。
当设置为低分辨率时，无论对方传真机使用什么分辨率与之握手，本端最终都固定使用低分辨率进行传输。
系统缺省的分辨率设置为低分辨率，

参　数： ChannelNo　　通道的逻辑号码。
FaxResolution 分辨率，取值如下：
FAX_RESOLUTION_LOW　　低分辨率
FAX_RESOLUTION_FINE　　高分辨率

返　回： 如果调用成功，返回 BF_OK

注　释： 系统缺省的分辨率为低分辨率。
该函数一定要在调用 BF_BindResource 函数之后调用，才能有效。

适用性： BF-PCI-V8F4、BF-PCI-V8F8

● BF_GetFaxResolution

原　型： unsigned short WINAPI BF_GetFaxResolution(unsigned short ChannelNo, unsigned short *FaxResolution);

功　能： 该函数获得传真的分辨率，该分辨率由函数 SetFaxResolution 设置，缺省分辨率为低分辨率，有高低两种。

参　数： ChannelNo　　通道的逻辑号码。
FaxResolution 分辨率，取值如下：
FAX_RESOLUTION_LOW　　低分辨率
FAX_RESOLUTION_FINE　　高分辨率

返　回： 如果调用成功，返回 BF_OK

注　释： 该函数一定要在调用 BF_BindResource 函数之后调用，才能有效。

适用性： BF-PCI-V8F4、BF-PCI-V8F8

● BF_SetFaxHeader

原　型： unsigned short WINAPI BF_SetFaxHeader(unsigned short ChannelNo,char *FaxFrom,char *FaxTo,char *FaxNo,char *FaxSubject,char *DateTime);

功　能： 该函数设置发送传真的传真头，即 From、To、Fax No、Subject 和 Date & Time 等。这些内容显示在发送传真每页的头部。式样如下：

To : xxxxxxxx From : xxxxxxxx Fax No : xxxxxxxx
Subject : xxxxxxxxxxxxxxxxxxxxxxxxxxxx Date : xxxxxxxx

该函数仅仅对四个发送传真函数起作用，分别是 BF_SendBitFaxFile、BF_SendMultiBitFaxFile、BF_SendBitFaxFileWithPageNo、BF_SendTiffFaxFile。

参　数：ChannelNo　通道的逻辑号码。
FaxFrom　发送传真方的号码或名称，最大字符数为 24，每个汉字算两个字符。
FaxTo　接收传真方的号码或名称，最大字符数为 24，每个汉字算两个字符。
FaxNo　发送传真方的传真机号码，最大字符数为 24，每个汉字算两个字符。
FaxSubject　传真主题，最大字符数为 55，每个汉字算两个字符。
DateTime　传真日期和时间，最大字符数为 24，每个汉字算两个字符。
格式可以为：2004 年 12 月 12 日　18 时 18 分。

返　回：如果调用成功，返回 BF_OK

注　释：这些内容共占两行，FaxFrom、FaxTo 和 FaxNo 放在同一行中，FaxSubject 和 DateTime 放在另一行。另外，该函数仅仅对四个发送传真函数起作用，分别是 BF_SendBitFaxFile、BF_SendMultiBitFaxFile、BF_SendBitFaxFileWithPageNo、BF_SendTiffFaxFile。

适用性：BF-PCI-V8F4、BF-PCI-V8F8

● BF_SetFaxSendHeaderFlag

原　型：unsigned short WINAPI BF_SetFaxSendHeaderFlag(unsigned short ChannelNo, unsigned short FaxSendHeaderFlag);

功　能：该函数设置在发送传真时，是否发送传真头信息，即由 BF_SetFaxHeader 函数设置的信息。

参　数：ChannelNo　语音通道号码，从 0 开始。
FaxSendHeaderFlag　是否发送传真头信息标志，取值如下：
0x0000 —— 不发送。
0x0001 —— 发送。

返　回：如果调用成功，返回 BF_OK

注　释：系统默认不发送。该函数一定要在 BF_SetFaxHeader 函数之前调用。
该函数一定要在调用 BF_BindResource 函数之后调用，才能有效。

适用性：BF-PCI-V8F4、BF-PCI-V8F8

● BF_SetFaxMaxRate

原　型：unsigned short WINAPI BF_SetFaxMaxRate(unsigned short ChannelNo, unsigned short FaxMaxRate);

功　能：该函数设置收发送传真的最大速率，该函数主要用于在条件恶劣的情况下，主动降低速度，以便成功收发传真。

参　数：ChannelNo　语音通道号码，从 0 开始。
FaxMaxRate　最大传真速率，建议取值如下：2400、4800、9600、14400。

返　回：如果调用成功，返回 BF_OK

注　释：该函数一定要在调用 BF_BindResource 函数之后调用，才能有效。

适用性：BF-PCI-V8F4、BF-PCI-V8F8

● BF_GetFaxInformation

原　型：unsigned short BF_GetFaxInformation(unsigned short ChannelNo, PBF_FAX_INFORMATION FaxInformation);

功　能：取得传真操作信息。

参　数：ChannelNo　语音通道号码，从 0 开始
FaxInformation　传真信息参数，是一个指向 PBF_FAX_INFORMATION 结构的指针，PBF_FAX_INFORMATION 的结构定义如下：

```
typedef struct _BF_FAX_INFORMATION{
    unsigned short SignalRate;
```

```
    unsigned short Resolution;
    unsigned short TwoDimensional;
    unsigned short RecordWidth;
    unsigned short RecordLength;
} BF_FAX_INFORMATION, *PBF_FAX_INFORMATION;
```

BF_FAX_INFORMATION 结构中的各项含义如下：

代号	含义
SignalRate	传真速率
Resolution	传真分辨率（0——3.58/mm，1——7.7l/mm）
TwoDimensional	是否二维编码（0——一维编码，1——二维编码）
RecordWidth	每行的点阵数量
RecordLength	每页的长度，单位毫米，取值如下： 0xFFFF ——长度不限 其他 ——单位为毫米的长度

返　回：　如果调用成功，返回 BF_OK

注　释：　该函数只有在得到事件 MESSAGE_FAX_HAVE_INFORMATION 后调用才有意义

适用性：　BF-PCI-V8F4、BF-PCI-V8F8

● BF_GetFaxPageNo

原　型：　unsigned short BF_GetFaxPageNo(unsigned short ChannelNo);

功　能：　获取当前正在进行的发送或接收传真的页数。

参　数：　ChannelNo　　语音通道号码，从 0 开始

返　回：　函数返回值为传真页数

注　释：　无

适用性：　BF-PCI-V8F4、BF-PCI-V8F8

● BF_SendFaxTxtBuffer

原　型：　unsigned short BF_SendFaxTxtBuffer(unsigned short ChannelNo, char *TxtBuffer, unsigned short BufferCount);

功　能：　发送格式为文本格式的缓存区传真。

参　数：　ChannelNo　　语音通道号码，从 0 开始
　　TxtBuffer　　文本格式的缓存区指针
　　BufferCount　　文本字节数量

返　回：　如果调用成功，返回 BF_OK

注　释：　在传真通道上返回事件 MESSAGE_SEND_FAX_OK 或 MESSAGE_SEND_FAX_ERROR
　　文本每行最大字节长度为 106 字节，并以 0x0A 为行结束。

适用性：　BF-PCI-V8F4、BF-PCI-V8F8

● BF_SendFaxTxtFile

原　型：　unsigned short BF_SendFaxTxtFile(unsigned short ChannelNo, char *FileName);

功　能：　发送格式为文本文件的传真。

参　数：　ChannelNo　　语音通道号码，从 0 开始
　　FileName　　文本文件名称

返　回：　如果调用成功，返回 BF_OK

注　释：　在传真通道上返回事件 MESSAGE_SEND_FAX_OK 或 MESSAGE_SEND_FAX_ERROR
　　文本每行最大字节长度为 106 字节，并以 0x0A 为行结束。

适用性：　BF-PCI-V8F4、BF-PCI-V8F8

● BF_SendBitFaxFile

原　型：unsigned short BF_SendBitFaxFile(unsigned short ChannelNo, char *FileName);

功　能：发送格式为 BitFax 文件的传真

参　数：ChannelNo　　语音通道号码，从 0 开始

FileName　　BitFax 文件名称

返　回：如果调用成功，返回 BF_OK

注　释：在传真通道上返回事件 MESSAGE_SEND_FAX_OK 或 MESSAGE_SEND_FAX_ERROR

适用性：BF-PCI-V8F4、BF-PCI-V8F8

● BF_SendBitFaxFileByPageNo

原　型：unsigned short WINAPI BF_SendBitFaxFileByPageNo(unsigned short ChannelNo,unsigned short StartPageNo,unsigned short EndPageNo, char *FileName);

功　能：该函数按照指定的页码发送传真。

参　数：ChannelNo　通道的逻辑号码。

StartPageNo　起始页码，从 1 开始，可以等于 EndPageNo，但不能大于。

EndPageNo　结束页码，可以等于 StartPageNo，也可以超过文件总的页码，超过时，发送到文件结束为止。

FileName　BitFax 格式传真文件名称。

返　回：如果调用成功，返回 BF_OK

注　释：无

适用性：BF-PCI-V8F4、BF-PCI-V8F8

● BF_SendMultiBitFaxFile

原　型：unsigned short WINAPI BF_SendMultiBitFaxFile(unsigned short ChannelNo, char *Directory,char *FileName);

功　能：该函数可以一次发送三个 BitFax 格式的文件。

参　数：ChannelNo　通道的逻辑号码。

Directory　传真文件的目录，对参数 FileName 中的所有文件都起作用，并可以为空(NULL)。其后不需要带“\”号。

FileName　BitFax 文件名称，最多可以三个文件名称，它们之间使用加号（+）相间隔，每个文件名称字节长度必须小于 0x80，否则该函数会返回失败

返　回：如果调用成功，返回 BF_OK，否则返回 BF_ERROR

注　释：无

适用性：BF-PCI-V8F4、BF-PCI-V8F8

● BF_SendTiffFaxFile

原　型：unsigned short BF_SendTiffFaxFile(unsigned short ChannelNo, char *FileName);

功　能：发送格式为 Tiff 文件的传真。

参　数：ChannelNo　　语音通道号码，从 0 开始

FileName　　Tiff Fax 格式文件名称

返　回：如果调用成功，返回 BF_OK

注　释：在传真通道上返回事件 MESSAGE_SEND_FAX_OK 或 MESSAGE_SEND_FAX_ERROR

适用性：BF-PCI-V8F4、BF-PCI-V8F8

● BF_SendTiffFaxFileWithPageNo

原　型：unsigned short WINAPI BF_SendTiffFaxFileWithPageNo(unsigned short ChannelNo, unsigned short StartPageNo, unsigned short EndPageNo, char *FileName);

功　能：　该函数按照指定的页码发送格式为 Tiff 的传真文件。

参　数：　ChannelNo　通道的逻辑号码。

StartPageNo　起始页码，从 1 开始，可以等于 EndPageNo，但不能大于。

EndPageNo　结束页码，可以等于 StartPageNo，也可以超过文件总的页码，超过时，发送到文件结束为止。

FileName　Tiff 格式传真文件名称。

返　回：　如果调用成功，返回 BF_OK

注　释：　无

适用性：　BF-PCI-V8F4、BF-PCI-V8F8

● BF_ReceiveBitFaxFile

原　型：　unsigned short BF_ReceiveBitFaxFile(unsigned short ChannelNo, char *FileName);

功　能：　接收传真，传真格式为 BitFax 格式。

参　数：　ChannelNo　语音通道号码，从 0 开始

FileName　存放 BitFax 格式传真数据的文件名称

返　回：　如果调用成功，返回 BF_OK

注　释：　在传真通道上返回事件 MESSAGE_RECEIVE_FAX_OK 或 MESSAGE_RECEIVE_FAX_ERROR

适用性：　BF-PCI-V8F4、BF-PCI-V8F8

● BF_ReceiveTiffFaxFile

原　型：　unsigned short BF_ReceiveTiffFaxFile(unsigned short ChannelNo, char *FileName);

功　能：　接收传真，传真格式为 Tiff 格式。

参　数：　ChannelNo　语音通道号码，从 0 开始

FileName　存放 Tiff Fax 格式传真数据的文件名称

返　回：　如果调用成功，返回 BF_OK

注　释：　在传真通道上返回事件 MESSAGE_RECEIVE_FAX_OK 或 MESSAGE_RECEIVE_FAX_ERROR

适用性：　BF-PCI-V8F4、BF-PCI-V8F8

● BF_GetFaxLength

原　型：　unsigned short WINAPI BF_GetFaxLength(unsigned short ChannelNo, unsigned long *FaxLength);

功　能：　获取发送或接收传真的字节长度。

参　数：　ChannelNo　语音通道号码，从 0 开始

FaxLength　发送或接收传真的字节长度，以页为单位

返　回：　如果调用成功，返回 BF_OK

注　释：　无

适用性：　BF-PCI-V8F4、BF-PCI-V8F8

● BF_SendMultiTiffFaxFile

原　型：　unsigned short WINAPI BF_SendMultiTiffFaxFile(unsigned short ChannelNo,PSEND_MULTI_FAX_INFO FileInfo,unsigned short FileCount)

功　能：　发送多个 TIFF 文件。

参　数：　ChannelNo　语音通道号码，从 0 开始

FileInfo　SEND_MULTI_FAX_INFO 结构数组。其结构如下：

```
typedef struct _SEND_MULTI_FAX_INFO
{
```

```
    char FileName[256];              // 传真文件名称
    unsigned short StartPage;        // 起始页码
    unsigned short EndPage;          // 结束页码
    unsigned short Resolution;       // 分辨率
    unsigned short Type;             // 文件操作类型
} SEND_MULTI_FAX_INFO, *PSEND_MULTI_FAX_INFO;
```

其中：文件操作类型，取值如下：

SEND_MULTI_FAX_FILE_TYPE_LAST 最后一个发送文件

SEND_MULTI_FAX_FILE_TYPE_VALID 当前文件被发送

SEND_MULTI_FAX_FILE_TYPE_PASS 忽略当前文件

FileCountSEND_MULTI_FAX_INFO 结构数组的数量。

返　回：　如果调用成功，返回 BF_OK

注　释：　无

适用性：　BF-PCI-V8F4、BF-PCI-V8F8

● BF_StopFaxOperate

原　型：　unsigned short BF_StopFaxOperate(unsigned short FaxChannelNo);

功　能：　停止传真操作。

参　数：　FaxChannelNo　　传真通道号码，从 0 开始

返　回：　如果调用成功，返回 BF_OK；但必须等待传真通道返回事件，才能表示传真结束。

注　释：　该函数仅仅停止相应传真通道的传真操作，并不释放与之关联的语音通道，如果需要释放连接的语音通道，必须调用函数 BF_FreeResource。

该函数一定要在调用 BF_FreeResource 函数之前调用，才能有效。

适用性：　BF-PCI-V8F4、BF-PCI-V8F8

第五部分 Digital 系列产品的技术支持

深圳市数灵通电子有限公司为 Digital 系列语音产品提供全面完善的售后服务和技术支持。

如果您在使用 Digital 系列语音产品时，遇到困难，可以遵循下列步骤处理：

(1) 仔细阅读相关产品用户手册的相关内容

(2) 检查您的源程序代码

(3) 参考 Digital 系列语音产品的 DEMO 例程

(4) 来电深圳市数灵通电子有限公司询问

深圳市数灵通电子有限公司将努力改进本公司的产品和用户手册，以使您的工作更省时省力。同时，我们也欢迎您对我们的产品和工作提出宝贵的意见，您可以使用您所喜欢的任何方式，在您方便的时候与我们联系，我们的联系方法是：

公司地址：深圳市南山区南新路锦福苑 7-1605

邮政编码：518033

联系电话：0755-26076088, 26076218

联系传真：0755-26076218

http://www.sltdigital.com